



**Universidad
Carlos III de Madrid**

ESCUELA POLITÉCNICA SUPERIOR

Grupo de Sistemas Electrónicos de Potencia (GSEP)

Departamento de Tecnología Electrónica

**DISEÑO, CONSTRUCCIÓN Y
VALIDACIÓN EXPERIMENTAL DE UN
INVERSOR MONOFÁSICO CON
CONTROL DIGITAL CONECTADO A
RED PARA PANELES FOTOVOLTAICOS**

Proyecto Fin de Carrera

INGENIERÍA INDUSTRIAL

AUTOR: LEANDRO E. BOYANO VANNUCCHI

TUTOR: PABLO ZÚMEL VAQUERO

JULIO 2009



CAPÍTULO 1. PLANTEAMIENTO Y OBJETIVOS DEL PROYECTO	9
CAPÍTULO 2. INTRODUCCIÓN.....	17
2.1. LA ENERGÍA SOLAR FOTOVOLTAICA	19
2.2. COMPONENTES DE LOS SISTEMAS FOTOVOLTAICOS	20
2.3. LA UNIDAD BÁSICA DE CONVERSIÓN (UBC)	22
CAPÍTULO 3. EL HARDWARE DEL SISTEMA	27
3.1. ESQUEMA FÍSICO DEL MONTAJE DEL PROTOTIPO DE INVERSOR + BLOQUE DE CONTROL	29
3.2. DISEÑO DEL INVERSOR MONOFÁSICO	30
3.2.1. CONSIDERACIONES DE DISEÑO.....	31
3.3. DISEÑO DEL SISTEMA DE MEDICIÓN Y CONTROL	37
3.3.1. EL BLOQUE DE SENSADO Y ACONDICIONAMIENTO.....	37
3.3.1.1. SENSADO DE TENSIÓN.	38
3.3.1.2. SENSADO DE CORRIENTE.....	43
3.3.2. EL CIRCUITO DE CONVERSIÓN DE DATOS.....	48
3.3.3. LA FPGA.....	51
3.3.4. FUENTES AUXILIARES.....	53
3.4. MONTAJE DEL PROTOTIPO DE PRUEBAS	54
CAPÍTULO 4. CONTROL DIGITAL DEL INVERSOR EN LAZO ABIERTO Y EN LAZO CERRADO CONECTADO A RED	59
4.1. EL CONTROL DEL PUENTE INVERSOR EN BUCLE ABIERTO. ...	62
4.1.1. SIMULACIÓN DEL BLOQUE DE DISPARO DEL INVERSOR EN BUCLE ABIERTO.....	65
4.1.2. VALIDACIÓN EXPERIMENTAL DEL BLOQUE DE DISPARO DEL INVERSOR EN BUCLE ABIERTO.	67
4.2. EL CONTROL DE LOS CONVERTIDORES A/D.	71
4.2.1. SIMULACIÓN DEL BLOQUE DE CONTROL PARA EL CONVERTIDOR A/D.....	75
4.2.2. VALIDACIÓN EXPERIMENTAL DEL BLOQUE DE CONTROL PARA EL CONVERTIDOR A/D.	78



4.3.	EL CONTROL DEL INVERSOR EN LAZO CERRADO CONECTADO A RED.	80
4.3.1.	EL LAZO DE CORRIENTE DE SALIDA.	82
4.3.2.	SIMULACIÓN DEL INVERSOR CON LAZO DE CORRIENTE CONTROLADO POR HISTÉRESIS.	86
4.4.	VALIDACIÓN EXPERIMENTAL DEL INVERSOR CON LAZO DE CORRIENTE CONECTADO A RED.	89
4.4.1.	MONITORIZACIÓN DE SEÑALES DIGITALES	89
4.4.2.	VALIDACIÓN EXPERIMENTAL DEL REGULADOR POR HISTÉRESIS CON CARGA INDUCTIVA A LA SALIDA DEL INVERSOR.	92
4.4.3.	VALIDACIÓN EXPERIMENTAL DEL REGULADOR POR HISTÉRESIS CON CARGA RESISTIVA A LA SALIDA DEL INVERSOR	94
4.4.4.	VALIDACIÓN EXPERIMENTAL DEL REGULADOR POR HISTÉRESIS CONECTADO DIRECTAMENTE A LA RED	97
4.5.	CONCLUSIONES	102
CAPÍTULO 5. PROPUESTA DE ALGORITMO DE MPPT SIN SENSORES DEL LADO DE CC.		103
5.1.	EL ALGORITMO DE CONTROL PARA EL SEGUIMIENTO DEL PUNTO DE MÁXIMA POTENCIA (MPPT).	105
5.2.	SIMULACIÓN DEL INVERSOR CON EL LAZO DE CORRIENTE Y EL CONTROL DEL MPPT PROPUESTO	110
5.3.	CONCLUSIONES	113
CAPÍTULO 6. ESTUDIO ECONÓMICO		115
6.1.	COSTES DE DESARROLLO	117
6.2.	COSTES DE LOS MATERIALES	118
6.3.	COSTE TOTAL DEL PROYECTO	118
6.4.	PRESUPUESTO DEL PROYECTO	119
CAPÍTULO 7. CONCLUSIONES Y TRABAJOS FUTUROS		121
7.1.	CONCLUSIONES	123
7.2.	TRABAJOS FUTUROS	125
BIBLIOGRAFÍA Y REFERENCIAS.		127



ANEXOS.	131
ANEXO I. ESQUEMÁTICO Y LAYOUT DE LA PCB DE LA ETAPA DE POTENCIA	133
ANEXO II. ESQUEMÁTICO Y LAYOUT DE LA PCB DE LOS CONVERTIDORES A/D	137
ANEXO III. CÁLCULOS CORRESPONDIENTES AL CIRCUITO DE ACONDICIONAMIENTO DE U_{RED} SENSADA	141
ANEXO IV. CÁLCULOS CORRESPONDIENTES AL CIRCUITO DE ACONDICIONAMIENTO DE I_0 SENSADA	143
ANEXO V. SCRIPT REALIZADO CON MATLAB PARA LA GENERACIÓN DE TABLAS DE DISPARO	145
ANEXO VI. DESCRIPCIÓN VHDL PARA LA GENERACIÓN DE UNA PWM UNIPOLAR	147
ANEXO VII. DESCRIPCIÓN VHDL PARA EL CONTROL DE LOS CONVERTIDORES A/D	167
ANEXO VIII. DESCRIPCIÓN VHDL PARA EL CONTROL DEL REGULADOR POR HISTÉRESIS	183
ANEXO IX. ENSAYOS EXPERIMENTALES DEL INVERSOR EN LAZO CERRADO	209
A. ENSAYOS EN CORTOCIRCUITO	209
B. ENSAYOS CON CARGA RESISTIVA	213
C. ENSAYOS DE CONEXIÓN A RED	216
ANEXO X. MODELO DEL PANEL FOTOVOLTAICO PARA SIMULACIONES CON PSIM	223
ANEXO XI. LISTA DE COMPONENTES EMPLEADOS	225



CAPÍTULO 1

<i>Figura 1.1. Esquema general del proyecto UNISOL. Diagrama de bloques.....</i>	<i>11</i>
<i>Figura 1.2. Esquema hardware del sistema inversor.....</i>	<i>14</i>

CAPÍTULO 2

<i>Figura 2.1. Panel solar</i>	<i>19</i>
<i>Figura 2.2. Componentes por bloques de la UBC del proyecto UNISOL.....</i>	<i>22</i>
<i>Figura 2.3. Núcleo magnético de la UBC del proyecto UNISOL</i>	<i>24</i>

CAPÍTULO 3

<i>Figura 3.1. Esquema de conexión físico de los diversos componentes del conjunto Inversor+Bloque de control</i>	<i>30</i>
<i>Figura 3.2. Conexión básico del inversor monofásico.....</i>	<i>31</i>
<i>Figura 3.3. Driver de disparo y elementos de protección para el disparo de los MOSFET's.....</i>	<i>33</i>
<i>Figura 3.4. Disipador empleado para las pérdidas de los MOSFET's.</i>	<i>35</i>
<i>Figura 3.5. PCB con la etapa de potencia y el bloque de sensado del prototipo de pruebas del inversor monofásico.....</i>	<i>36</i>
<i>Figura 3.6. Layout de la PCB con los distintos planos de masa empleados</i>	<i>36</i>
<i>Figura 3.7. Diagrama de bloques del sistema de control.</i>	<i>37</i>
<i>Figura 3.8. Dispositivo transductor de tensión LV 25-P.</i>	<i>39</i>
<i>Figura 3.9. Circuito de conexión del sensor de efecto Hall transductor de tensión.....</i>	<i>39</i>
<i>Figura 3.10. Tensión de red sensada antes y después del circuito de acondicionamiento</i>	<i>41</i>
<i>Figura 3.11. Circuito de acondicionamiento del sensor de tensión.....</i>	<i>41</i>
<i>Figura 3.12. Equivalente de Thevenin para el circuito de acondicionamiento de la tensión sensada.....</i>	<i>42</i>
<i>Figura 3.13. Dispositivo transductor de corriente LTS 25-NP.....</i>	<i>43</i>
<i>Figura 3.14. Característica de funcionamiento del sensor de corriente. Curva de tensión de salida (V_{OUT}) frente a corriente medida (I_P)</i>	<i>44</i>
<i>Figura 3.15. Buffer de entrada del circuito convertidor A/D.</i>	<i>45</i>
<i>Figura 3.16. Modificación del buffer de entrada para el circuito de acondicionamiento de la corriente sensada.....</i>	<i>46</i>
<i>Figura 3.17. Circuito de acondicionamiento de la corriente i_O sensada.....</i>	<i>47</i>
<i>Figura 3.18. Corriente entregada a la red, sensada antes y después del circuito de acondicionamiento.</i>	<i>47</i>
<i>Figura 3.19. Circuito de conversión de señales analógicas a digitales.</i>	<i>48</i>
<i>Figura 3.20. El convertidor A/D modelo ADS7887.</i>	<i>49</i>
<i>Figura 3.21. Cronograma de tiempos para el funcionamiento del convertidor ADS7887.</i>	<i>50</i>
<i>Figura 3.22. Esquema eléctrico del convertidor A/D.</i>	<i>50</i>
<i>Figura 3.23. Tarjeta de evaluación FPGA empleada para el control del inversor y los ADC.</i>	<i>52</i>
<i>Figura 3.24. Fuentes de alimentación auxiliares.....</i>	<i>54</i>



Figura 3.25. Conexionado entre la tarjeta de evaluación de la FPGA y la tarjeta de conversión de los ADC's.	55
Figura 3.26. Cables de conexión entre los diferentes componentes del prototipo de pruebas.	57
Figura 3.27. Montaje completo del prototipo de pruebas diseñado para el inversor monofásico del proyecto UNISOL.	57

CAPÍTULO 4

Figura 4.1. Situación del hardware digital dentro del diagrama de bloques del sistema completo.	61
Figura 4.2. Esquemático del bloque de control del puente inversor en bucle abierto sintetizado en la FPGA.	62
Figura 4.3. Representación de la PWM obtenida a partir de la comparación del contador de red y los ángulos de corte.	63
Figura 4.4. Esquema del proceso de reprogramación de la FPGA con la PWM unipolar.	64
Figura 4.5. Banco de pruebas empleado para la simulación del inversor en lazo abierto.	66
Figura 4.6. Simulación en VHDL del control del puente inversor en lazo abierto con frecuencias de conmutación de 1.5 kHz, 3 kHz y 10 kHz.	67
Figura 4.7. Fuente de alimentación modelo 6015A, de HP/Agilent.	68
Figura 4.8. Corriente de salida del inversor monofásico en bucle abierto, empleando $F_{CONMUTACIÓN} = 30 \text{ kHz}$, $m_a = 0.8$ y $t_{MUERTO} = 1 \mu s$	69
Figura 4.9. Diagrama de bloques del prototipo de pruebas con esquema básico del hardware digital.	72
Figura 4.10. Cronograma de tiempos para el funcionamiento del convertidor ADS7887.	72
Figura 4.11. Implementación del bloque de control de los ADC's y su conexionado básico.	74
Figura 4.12. Máquina de estados para el control de los convertidores A/D.	75
Figura 4.13. Banco de pruebas para la simulación del hardware digital del control de los ADC's.	76
Figura 4.14. Simulación en VHDL del bloque de control para el convertidor A/D.	77
Figura 4.15. Montaje realizado para la validación experimental del hardware digital de control de la tarjeta de los ADC's.	78
Figura 4.16. Validación experimental del control del convertidor A/D mediante una señal triangular de 2 kHz y 4.4 Vpico.	80
Figura 4.17. Diagrama de bloques del inversor completo, con lazo de corriente y seguimiento del punto de máxima potencia.	81
Figura 4.18. Diagrama de bloques del inversor con el lazo de corriente.	83
Figura 4.19. Representación del funcionamiento del regulador de corriente por histéresis.	84
Figura 4.20. Diagrama de estados del regulador por histéresis.	84
Figura 4.21. Diagrama de bloques implementado en la FPGA para el regulador por histéresis.	85



<i>Figura 4.22. Banco de ensayos creado para la simulación del inversor con regulador por histéresis.</i>	<i>87</i>
<i>Figura 4.23. Simulación en MODELSIM del inversor con el regulador de corriente por histéresis.</i>	<i>88</i>
<i>Figura 4.24. Convertidor D/A para la medición de señales digitales de la FPGA.</i>	<i>90</i>
<i>Figura 4.25. Esquema de montaje de los convertidores D/A.</i>	<i>91</i>
<i>Figura 4.26. Tensión de red sensada y señal de tensión de red digitalizada y pasada por el convertidor D/A.</i>	<i>91</i>
<i>Figura 4.27. Esquema eléctrico empleado para la validación experimental del inversor en cortocircuito funcionando en lazo cerrado.</i>	<i>92</i>
<i>Figura 4.28. Validación experimental del control del inversor en lazo cerrado con carga inductiva, tensión de panel de 100 V y corriente de referencia de 2 A de pico... 94</i>	
<i>Figura 4.29. Esquema eléctrico empleado para la validación experimental del inversor con carga resistiva de 106 Ω funcionando en lazo cerrado.</i>	<i>95</i>
<i>Figura 4.30. Validación experimental del control del inversor en lazo cerrado con carga resistiva ($R=106 \Omega$), tensión de panel de 240 V y corriente de referencia de 2 A_{PICO}.</i>	<i>96</i>
<i>Figura 4.31. Imagen en detalle de la tensión y corriente de salida del inversor para unas condiciones de funcionamiento siguientes: Tensión de panel de 240 V y corriente de referencia de 2 A de pico.</i>	<i>97</i>
<i>Figura 4.32. Esquema eléctrico empleado para la validación experimental del inversor conectado a red.</i>	<i>98</i>
<i>Figura 4.33. Dispositivos empleados para la validación experimental del inversor en lazo cerrado conectado a red.</i>	<i>99</i>
<i>Figura 4.34. Filtro en T empleado para los ensayos del inversor conectado a red.</i>	<i>99</i>
<i>Figura 4.35. Resultado del ensayo referente al lazo de corriente del inversor con filtro LCL. Condiciones de funcionamiento: $U_{RED}=110 V_{ef}$, $I_{REF}=2 A_{pico}$.</i>	<i>100</i>

CAPÍTULO 5

<i>Figura 5.1. Diagrama de bloques del inversor en lazo cerrado con control MPPT. ...</i>	<i>106</i>
<i>Figura 5.2. Curvas de corriente y potencia frente a tensión en un panel solar fotovoltaico.</i>	<i>107</i>
<i>Figura 5.3. Flujograma correspondiente al control del MPPT.</i>	<i>109</i>
<i>Figura 5.4. Esquemático empleado para la simulación en PSIM del control del inversor en lazo cerrado con MPPT.</i>	<i>111</i>
<i>Figura 5.5. Resultado de la simulación en PSIM del control del MPPT.</i>	<i>113</i>



CAPÍTULO 1.

PLANTEAMIENTO Y OBJETIVOS DEL PROYECTO

El presente proyecto forma parte de un proyecto de mayor amplitud denominado UNISOL, desarrollado íntegramente por el Grupo de Sistemas Electrónicos de Potencia (GSEP), perteneciente al Departamento de Tecnología Electrónica de la Universidad Carlos III. El proyecto UNISOL es la continuación de un proyecto anterior realizado por el GSEP, denominado proyecto AGENDA, y a su vez, es el antecesor al denominado proyecto INAFSOL.

Estos tres proyectos comparten un objetivo común, que consiste en captar la energía solar incidente sobre una serie de paneles solares fotovoltaicos, para convertirla en energía eléctrica, que es convenientemente tratada para ser inyectada a la red eléctrica en forma de corriente alterna. Para ello, se ha desarrollado en cada uno de los proyectos, una Unidad Básica de Conversión de Potencia (UBC), formada principalmente por un inversor, que permite conectar los paneles fotovoltaicos a la red, controlando y adaptando la inyección de energía. La UBC es un elemento orientado principalmente a la generación de energía distribuida, por lo que ha de cumplir una serie de premisas importantes tales como la modularidad y el tamaño reducido.

En la *Figura 1.1* se presenta un esquema general por bloques del proyecto UNISOL, centrado principalmente en el desarrollo de la UBC.

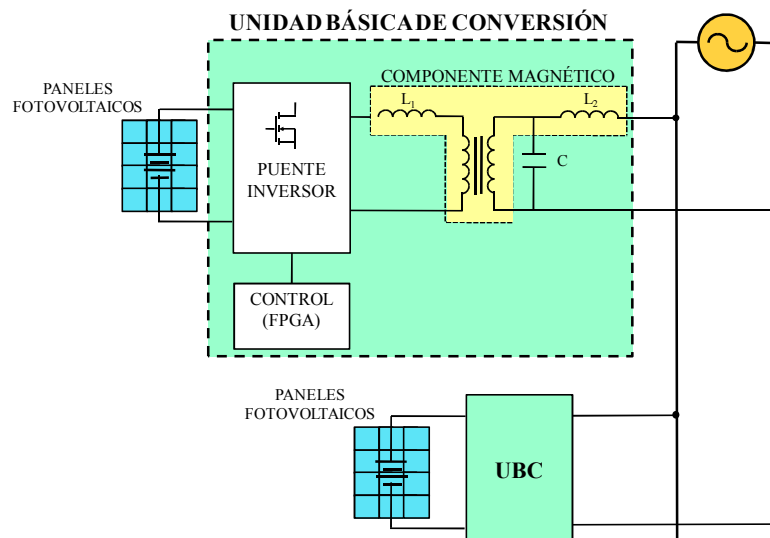


Figura 1.1. Esquema general del proyecto UNISOL. Diagrama de bloques.



A continuación se detallan a grandes rasgos las diferencias entre los tres proyectos mencionados anteriormente:

En el proyecto AGENDA, la UBC está constituida por un convertidor CC/CC para el seguimiento del punto de máxima potencia (MPPT), junto con una etapa inversora monofásica, gobernados independientemente mediante control digital. La diferencia principal del proyecto UNISOL radica en que, mientras que en el proyecto AGENDA se utilizaba un convertidor CC/CC para el MPPT, en el proyecto UNISOL se posibilita la integración de la parte del MPPT en el diseño del control digital del inversor, permitiendo el funcionamiento de los paneles fotovoltaicos en su punto de máxima potencia. Por su parte, el proyecto INAFSOL basa su UBC en un inversor trifásico, logrando así un aumento de potencia entregada a la red con igualdad de tamaño de los componentes magnéticos.

El proyecto UNISOL tiene además la singularidad de tratarse de una de las primeras experiencias del GSEP con inversores en lazo cerrado conectados a la red eléctrica.

Para el desarrollo de la UBC del proyecto UNISOL se ha diseñado un inversor monofásico, en el cual se han implementado técnicas de control que cumplen al mismo tiempo un doble objetivo: la correcta conexión y sincronización con la red eléctrica y la imposición, en su entrada, de una tensión continua que posibilita el funcionamiento de los paneles fotovoltaicos en el punto de máxima potencia. Además, se ha realizado un procedimiento de diseño para la integración de componentes magnéticos, que permite la obtención de un único componente que integra las funciones de filtro, transformador e inductancia de conexión a red. Finalmente, se ha adaptado el diseño del equipo de acuerdo a las Normas en vigor dentro de la UE. Para ello se ha diseñado un filtro EMI de red, con topología de doble etapa.

Este proyecto de fin de carrera se centra en la parte de la UBC correspondiente a la realización del diseño y montaje de una etapa inversora monofásica, así como el control digital de la misma, con el fin de transformar la



energía recogida por los paneles fotovoltaicos en corriente alterna que pueda ser inyectada a la red eléctrica.

Todos los resultados obtenidos, tanto en bucle abierto como en lazo cerrado, han sido simulados en primera instancia con la herramienta de simulación de sistemas digitales basados en lenguaje VHDL, MODELSIM. A su vez, se han validado experimentalmente todos los resultados con un prototipo de pruebas diseñado a tal efecto. Este prototipo está formado por varios elementos que se detallan a continuación:

- Se ha diseñado un circuito impreso (PCB) donde se integra la electrónica de potencia correspondiente a la etapa inversora, conjuntamente con la electrónica necesaria para el sensado de los parámetros eléctricos más importantes a medir en el sistema.
- Se ha implementado otra PCB con varios convertidores A/D que tienen la misión de transformar las señales analógicas captadas por los sensores en señales digitales que sean procesables por el sistema de control digital.
- El sistema de control digital está constituido por un kit básico de evaluación para FPGA's, concretamente el modelo BASYS de la compañía DIGILENT INC., en cuyo diseño alberga una FPGA modelo SPARTAN 3E de XILINX. Este dispositivo es el encargado de capturar los valores de las señales eléctricas digitalizadas por los ADC y de generar a su vez los disparos del inversor.

En la *Figura 1.2* se puede observar un esquema del hardware del sistema inversor con todos los componentes de la UBC, así como un panorama más detallado del sistema de control realizado en este proyecto.

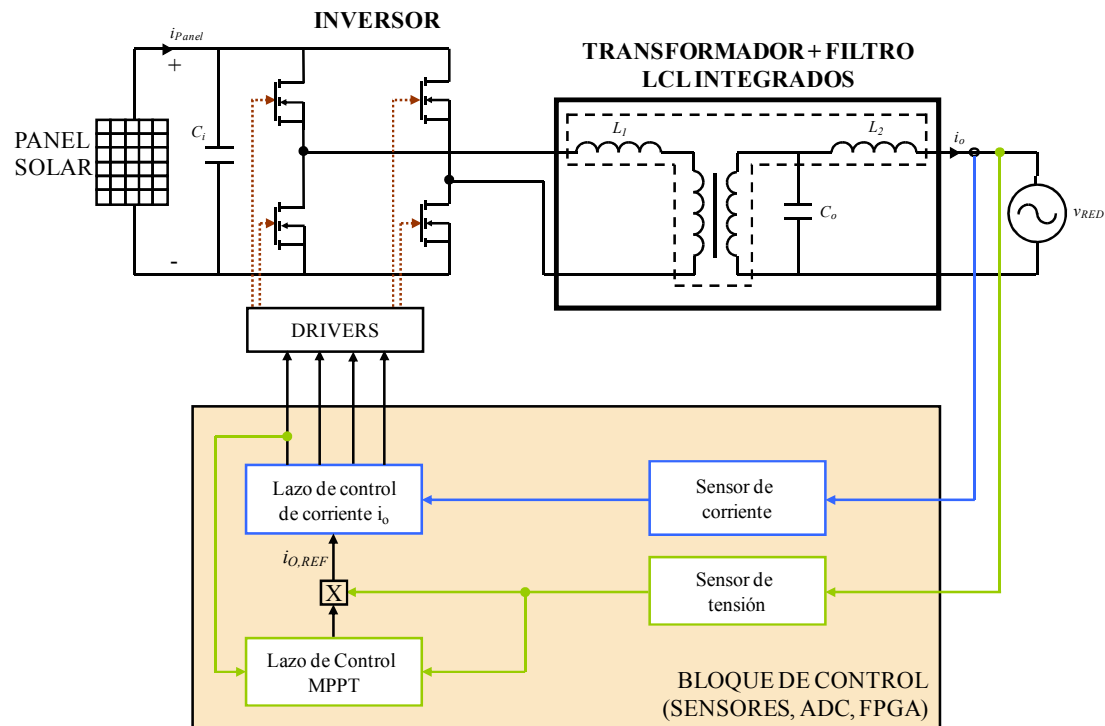


Figura 1.2. Esquema hardware del sistema inversor.

El diseño del circuito de control digital que se ha implementado para el inversor de este proyecto está compuesto por 2 lazos:

- El primero de ellos es un lazo de control de la corriente de salida. Su diseño está basado en un regulador por histéresis que genera directamente los pulsos para el disparo de los transistores.
- El otro lazo se encarga de generar la corriente de referencia del lazo de corriente para obtener la máxima potencia del panel fotovoltaico.

El objetivo principal de este proyecto es **construir un inversor monofásico para inyectar potencia a la red eléctrica, proveniente de una fuente de corriente continua (típicamente un panel solar fotovoltaico)**. Este objetivo se puede dividir en los siguientes objetivos parciales:

- Diseñar y construir un inversor monofásico con la circuitería auxiliar (drivers y fuentes de alimentación) para poder funcionar conectado únicamente a la red.



- Desarrollar e implementar el control del inversor para poder inyectar potencia a la red (lazo de corriente).
- Diseñar y construir los circuitos de sensado, acondicionamiento y conversión de datos necesarios para lograr el objetivo anterior.
- Proponer un algoritmo MPPT con un uso mínimo de sensores para el funcionamiento del inversor conectado a red.



CAPÍTULO 2.

INTRODUCCIÓN

En este capítulo introductorio se realiza una descripción general de los componentes de un sistema de transferencia de energía solar desde que es recogida por las células fotovoltaicas de los paneles solares fotovoltaicos hasta que es transmitida a la red eléctrica, haciendo especial mención a la parte de transformación de energía eléctrica en corriente continua (así es como se recoge en los paneles fotovoltaicos) a una corriente alterna sinusoidal de 50 Hz, que es la forma de transmisión a la red eléctrica en Europa.

2.1. LA ENERGÍA SOLAR FOTOVOLTAICA

Se denomina energía solar fotovoltaica a una forma de obtención de energía eléctrica a través de paneles fotovoltaicos. Ver *Figura 2.1*.



Figura 2.1. Panel solar

Los paneles fotovoltaicos están formados por dispositivos semiconductores tipo diodo que, al recibir radiación solar, se excitan y provocan saltos electrónicos, generando una pequeña diferencia de potencial en sus extremos. Estos elementos se llaman células fotovoltaicas, son de silicio, y se pueden acoplar en serie permitiendo obtener mayores voltajes. [www.wikipedia.org]



2.2. COMPONENTES DE LOS SISTEMAS FOTOVOLTAICOS

Los componentes de un sistema fotovoltaico dependen del tipo de aplicación que se considere y de las características de la instalación. En caso de tratarse de un sistema autónomo, los componentes necesarios para que funcione correctamente y tenga una elevada fiabilidad son: placas fotovoltaicas, acumulador eléctrico, regulador de carga e inversor. En cambio, las instalaciones conectadas a la red eléctrica se caracterizan por no incorporar acumuladores, ya que la energía que se envía a la red no necesita acumularse, sino que se emplea únicamente una unidad básica de conversión que adecua la corriente para ser inyectada a la red.

LAS PLACAS FOTOVOLTAICAS

El principal elemento de una instalación fotovoltaica son las placas o paneles solares, constituidos por un conjunto de células fotovoltaicas capaces de captar la energía solar y transformarla en energía eléctrica en forma de corriente continua.

EL ACUMULADOR ELÉCTRICO

La función prioritaria de los acumuladores eléctricos o baterías en un sistema de generación fotovoltaico es la de acumular la energía que se produce durante las horas de luminosidad para poder ser utilizada en la noche o durante periodos prolongados de falta de luz. La utilización de acumuladores está motivada por el hecho de que la intensidad solar varía a lo largo del día y del año y así varía también la producción de electricidad del sistema. Sin embargo, las necesidades de las personas no varían en ese mismo orden, y por tanto, el sistema debe acumular energía de manera constante para cuando no la tenga disponible. Con este fin se utilizan generalmente acumuladores eléctricos, que es el sistema más económico y eficiente de que se dispone.

Otra función importante de las baterías es la de proveer, durante cortos periodos de tiempo, una intensidad de corriente superior a la que el dispositivo fotovoltaico puede entregar. Tal es el caso de un motor, que en el momento del arranque puede



demandar una corriente de 4 a 6 veces su corriente nominal durante unos pocos segundos.

EL REGULADOR DE CARGA

Es el dispositivo encargado de proteger a la batería frente a sobrecargas y sobredescargas profundas. El regulador de tensión controla constantemente el estado de carga de las baterías y regula la intensidad de carga de las mismas para alargar su vida útil. También genera alarmas en función del estado de dicha carga. Los reguladores actuales contienen microcontroladores para la correcta gestión del sistema fotovoltaico. Su programación elaborada permite un control capaz de adaptarse a las distintas situaciones de forma automática, permitiendo la modificación manual de sus parámetros de funcionamiento para instalaciones especiales.

Existen dos tipos de reguladores de carga: los reguladores lineales y los conmutados.

EL INVERSOR

El circuito inversor tiene como objetivo convertir la corriente continua (CC) de la instalación fotovoltaica en corriente alterna (CA). Su principio de funcionamiento se basa en el empleo de transistores que actúan a modo de interruptores permitiendo interrumpir las corrientes e invertir su polaridad.

El inversor tiene dos posibles funcionalidades:

- Utilizar cargas de CA en instalaciones aisladas de la red.
- Conectar los sistemas fotovoltaicos a la red de distribución eléctrica.

2.3. LA UNIDAD BÁSICA DE CONVERSIÓN (UBC)¹

La UBC es la denominación que se da en este proyecto al conjunto de elementos que forman parte del módulo de conversión de la energía por el que se extrae la energía de los paneles fotovoltaicos y se introduce en la red eléctrica (ver *Figura 2.2*). Estos elementos se enumeran a continuación

- Un inversor monofásico controlado digitalmente mediante una FPGA.
- Un circuito magnético que integra un filtro LCL y un transformador.
- Un Filtro EMI de red con topología de doble etapa.

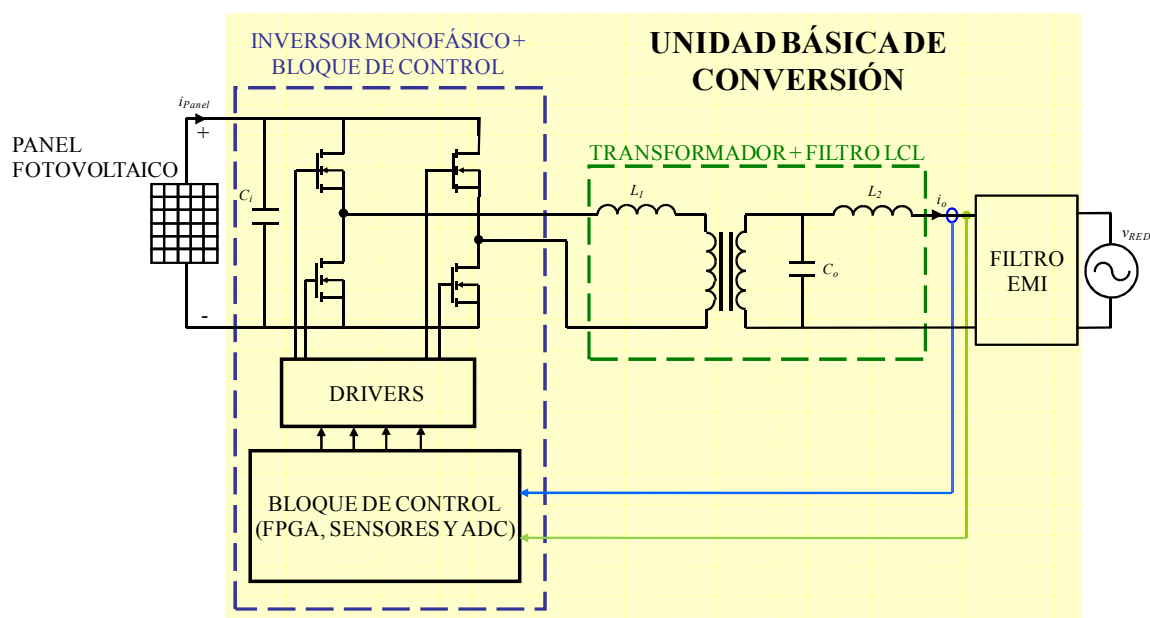


Figura 2.2. Componentes por bloques de la UBC del proyecto UNISOL.

La UBC es un elemento destinado a controlar y adaptar la inyección de energía eléctrica, procedente de paneles solares fotovoltaicos, a la red de distribución eléctrica.

El proyecto UNISOL se centra en el desarrollo de una UBC orientada fundamentalmente a conseguir sistemas basados en el concepto de generación distribuida (ver *Figura 1.1*). Para ello, se han definido en primer lugar los criterios que

¹ La información de este apartado ha sido extraída del informe final del proyecto UNISOL (CP06: Inversor de Etapa Única con MPPT para conexión a Red de Sistemas Fotovoltaicos)



han de cumplirse, que son los siguientes: modularidad, escalabilidad (de estructura escalable), susceptible a economías de escala, de fácil mantenimiento y coste reducido.

A continuación se describen brevemente cada uno de los distintos elementos que forman parte de la UBC desarrollada en el proyecto UNISOL.

EL INVERSOR MONOFÁSICO

El inversor es el convertidor de potencia que se encarga de acondicionar la energía para adecuarla a la red y de extraer la máxima potencia del generador fotovoltaico. Desde el punto de vista topológico, se trata de un inversor en puente completo, utilizando MOSFET como interruptores.

EL SISTEMA DE CONTROL

El bloque de control mide las principales magnitudes eléctricas del sistema y genera los pulsos de control de los transistores del inversor, permitiendo el funcionamiento del sistema en lazo cerrado.

En este bloque se incluye el sistema de medida de las magnitudes eléctricas (sensores y su correspondiente acondicionamiento), los convertidores analógico-digitales y la FPGA, que integra los algoritmos de control del lazo de corriente y del MPPT.

EL TRANSFORMADOR Y FILTRO INTEGRADOS

La unidad básica de conexión entre el inversor y la red consta de un filtro tipo LCL y un transformador (Ver *Figura 2.2*).

El filtro LCL permite filtrar los armónicos de alta frecuencia generados a la salida del inversor y por tanto disponer de una corriente de salida filtrada. De esta forma, es posible realizar la transmisión de potencia a un sistema de tensión determinado, como es la red eléctrica, con un bajo contenido de armónicos. El transformador, por su parte, permite adaptar los niveles de tensión a la salida del sistema, al tiempo que proporciona un aislamiento galvánico.

En el proyecto UNISOL, se ha desarrollado una estrategia de diseño que permite integrar las dos inductancias y el transformador en un único componente. Esto permite reducir el peso y tamaño del conjunto, y posibilita un incremento de eficiencia energética.

En la *Figura 2.3* se muestra un esquema general del prototipo de núcleo magnético diseñado.

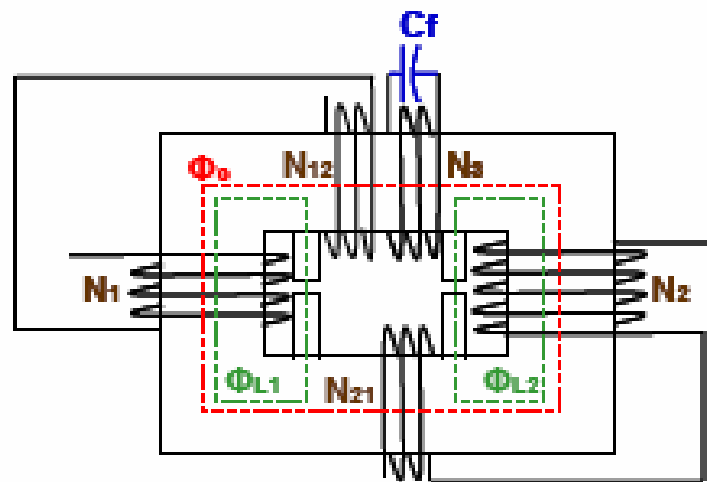


Figura 2.3. Núcleo magnético de la UBC del proyecto UNISOL

Las especificaciones técnicas del componente magnético diseñado en el proyecto UNISOL son las siguientes:

- Relación entre la tensión de salida y la de entrada 1:2.
- Atenuación del armónico de corriente a frecuencia de conmutación del inversor (máxima de aproximadamente 30 kHz) de al menos 30 dB.
- Potencia a entregar de 1 kW a 220 V_{rms}.
- Condensador de 1.5 μ F.

EL FILTRO EMI

Aunque en un principio en el proyecto UNISOL solo se plantea el diseño y construcción de un prototipo de pruebas, se pretende cumplir con la normativa vigente sobre Compatibilidad Electromagnética para cualquier equipo electrónico que vaya a ser



comercializado en la UE (*Normas EN 55011/22*). Por esta razón, y dado que los convertidores CC/CA son un caso particularmente difícil debido a la alta frecuencia de conmutación de los transistores, se han empleado 2 técnicas que logren disminuir el ruido EMI a valores aceptables. Estas técnicas son:

- En primer lugar se ha optimizado el *layout* del circuito impreso respecto a la posición física de los componentes y el trazado de pistas sobre el mismo. Para ello, se ha dedicado especial atención a la adecuada colocación de componentes críticos como los transistores MOSFET de potencia, la reducción de bucles donde circulan corrientes elevadas; y las conexiones largas de tierra.
- La segunda técnica ha consistido en construir un filtro de red específico con dos etapas y alta atenuación, diseñado con una metodología propia desarrollada por el equipo del proyecto.

Los resultados que se obtienen de la realización de estas técnicas implican una reducción del ruido EMI de 15 dB en la banda de frecuencias de entre 0.15 y 30 MHz. No obstante, no se ha logrado el objetivo de reducir los niveles por debajo de los límites que marca la UE, debido a que el nivel de ruido EMI generado por el prototipo ha resultado ser muy elevado, del orden de 40 dB por encima del límite máximo permitido. De este modo, se hace evidente la dificultad de construir un prototipo que cumpla los objetivos de funcionalidad y compatibilidad electromagnética.



CAPÍTULO 3.

EL HARDWARE DEL SISTEMA



En este capítulo se describe el proceso de diseño y montaje de todos los componentes hardware que han sido necesarios para la creación del prototipo de pruebas del inversor del proyecto UNISOL. En este aspecto, se hace referencia a dos partes claramente diferenciadas. Por una parte se encuentra toda la circuitería destinada a la circulación de elevadas corrientes, denominada comúnmente como etapa de potencia, que está constituida principalmente por el puente inversor. Por otro lado está la parte dedicada al control o circuitería de señal, por donde circulan corrientes pequeñas en comparación con la etapa de potencia. Dentro del hardware destinado al control del sistema se describen a su vez 3 partes: un bloque de sensado y acondicionamiento de las principales señales analógicas medidas en la etapa inversora; un bloque de conversión de los datos analógicos a digitales; y por último, una unidad de control digital basada en FPGA, que trata y gestiona los datos recibidos para realizar finalmente la generación de los disparos del inversor monofásico.

3.1. ESQUEMA FÍSICO DEL MONTAJE DEL PROTOTIPO DE INVERSOR + BLOQUE DE CONTROL

La realización de este proyecto ha comenzado con la implementación física de una serie de componentes que forman parte de la etapa inversora, así como del sensado y el control digital del prototipo de pruebas.

En la *Figura 3.1* se muestra un esquema detallado de las conexiones y los bloques físicos en los que está subdividida la construcción del prototipo de pruebas. En este esquema se observa que el diseño del inversor está constituido por 3 elementos bien diferenciados físicamente:

- En la denominada PCB1, se incluyen el inversor, junto con los sensores del sistema de control y el circuito de acondicionamiento de los sensores de tensión.
- La PCB2 está constituida por el circuito de conversión de los ADC's junto con el circuito de acondicionamiento de los sensores de corriente.
- Por otra parte está la FPGA, encargada de generar los disparos del inversor y de gestionar el control de los ADC's.

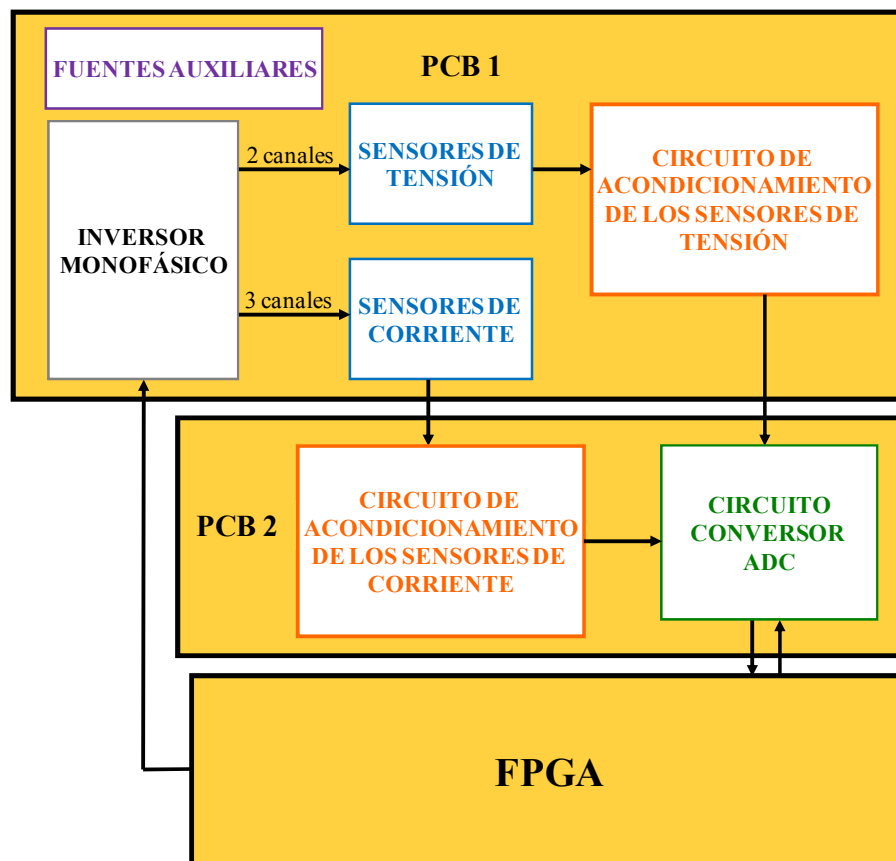


Figura 3.1. Esquema de conexionado físico de los diversos componentes del conjunto Inversor+Bloque de control

3.2. DISEÑO DEL INVERSOR MONOFÁSICO

El inversor utilizado para la realización del prototipo de pruebas es de tipo monofásico en puente completo. Su misión es la de convertir una tensión continua mediante unos transistores y un filtro, en una tensión sinusoidal a una frecuencia determinada en función de las necesidades del sistema. En este proyecto, el objetivo es conectar el convertidor a la red eléctrica, por lo que la tensión de salida ha de generarse a una frecuencia de 50 Hz; y además ha de estar sincronizada con la red.

El circuito inversor del proyecto UNISOL emplea como fuente de continua los paneles fotovoltaicos, que son los encargados de suministrar la tensión continua de entrada, mientras que a la salida del mismo está conectada el filtro LCL más transformador, que ofrecen el aislamiento galvánico y la inductancia necesaria para la

conexión a red. En la *Figura 3.2* se representa un diagrama de bloques donde se puede apreciar el conexionado básico del inversor.

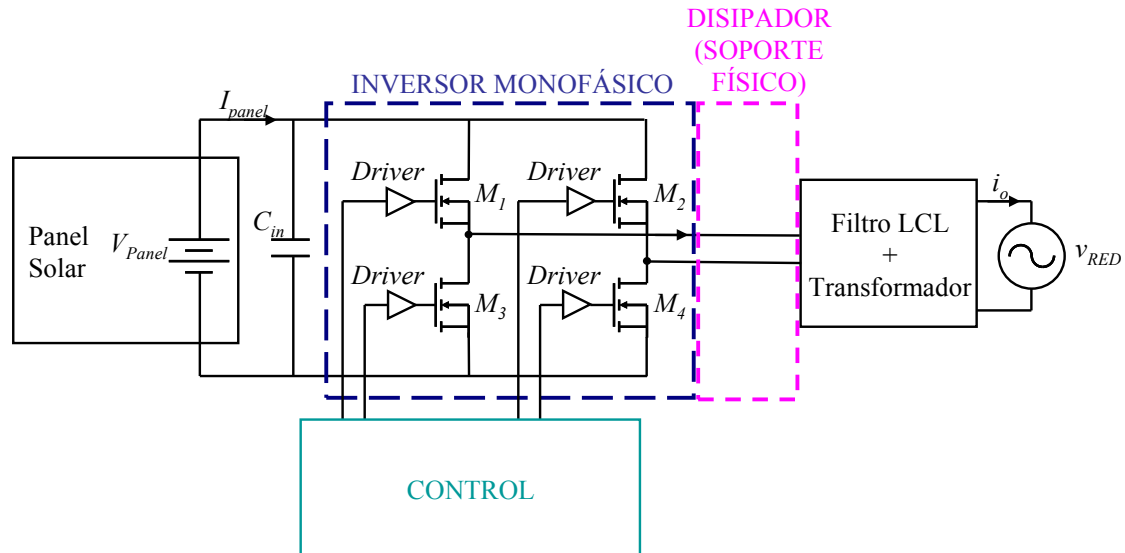


Figura 3.2. Conexionado básico del inversor monofásico.

Para realizar el diseño de la etapa inversora, se ha empleado la herramienta de simulación de circuitos electrónicos y diseño de circuitos impresos, ORCAD. En el *ANEXO I* se muestran tanto el esquemático como el layout utilizados para llevar a cabo el montaje del inversor.

3.2.1. CONSIDERACIONES DE DISEÑO

El primer paso a la hora de construir un inversor es conocer las especificaciones de diseño del mismo. Para la realización de este proyecto se han tenido en cuenta 2 posibles especificaciones de diseño. La primera de ellas corresponde a la especificación de diseño inicial para la que se concibió el proyecto, consistente en una tensión de salida del inversor de 110 V_{ef}, que después del transformador serían 220 V_{ef} de red. La otra especificación es una ampliación para poder emplear el inversor sin transformador, es decir, imponiendo 220 V_{ef} a la salida del inversor. En última instancia, se propone una especificación de diseño particularizada para la validación experimental del puente inversor funcionando en lazo cerrado, cuyos requerimientos máximos son inferiores que en el resto de consideraciones de diseño.



	Condición inicial	Ampliación	Pruebas
V_{AC} (a la salida del inversor)	110 V_{ef}	220 V_{ef}	110V_{ef} (nom) 210 V_{ef} (max)
V _{DC max}	175 V _{DC}	350 V _{DC}	315 V _{DC}
V _{AC max}	155 V _{pico}	311 V _{pico}	155 V _{pico} 300V _{pico}
I _{AC max}	10 A _{ef} (13 A _{pico})	5 A _{ef} (7 A _{pico})	1.4 A _{ef} (2 A _{pico})
I _{DC max}	7 A _{DC}	3.5 A _{DC}	<1 A
I _{TRANSISTOR MEDIA MAX}	6.5 A	3.5 A	<2 A
P _{max}	1 kW	1 kW	200 W
Comentario	Con el transformador integrado elevador 1:2, equivale a una tensión de red de 220 V _{ef}		Sensor ajustado para rango de corrientes de 2 A _{PICO}

Los componentes utilizados para el diseño de la etapa de potencia así como el bloque de sensado, se han escogido atendiendo a estas consideraciones de diseño, como se describe a continuación:

A. PUENTE INVERSOR (TRANSISTORES DE POTENCIA)

En este proyecto, el diseño propuesto para el prototipo de la UBC se basa en un inversor de una fase capaz de entregar una potencia de 1 kW. Para esta característica de diseño, se han empleado unos transistores MOSFET IRF740, capaces de soportar tensiones drenador-fuente de hasta 400 V y corrientes máximas por el drenador de 10 A. Según las especificaciones de diseño, estos transistores cumplen con los requerimientos máximos de 175 V máximos de continua y los 6.5 A que han de circular por cada uno de ellos en condiciones nominales.

B. CIRCUITOS DE DISPARO (DRIVERS) Y ELEMENTOS DE PROTECCIÓN PARA EL DISPARO DE LOS TRANSISTORES

Cada rama de MOSFET's lleva asociado un driver IR2110 que permite convertir los 3.3 V que salen del bloque de control digital en los 15 V necesarios para el disparo

de los transistores. El IR2110 se caracteriza por generar disparos para una rama del inversor, que incluye el disparo referido a una masa flotante (M_1 para la rama izquierda del inversor en la *Figura 3.3*). Además, estos drivers tienen la característica de soportar tensiones en cada rama de hasta $500 V_{DC}$, valor suficientemente elevado para este prototipo, si se tiene en cuenta que según especificaciones de diseño, no se superan los $175 V_{DC}$ a la entrada del inversor.

Otro elemento de vital importancia en este diseño es la inclusión de una resistencia en la puerta de cada MOSFET de 100Ω y $2W$, cuyo objetivo es impedir que circule una corriente demasiado elevada por la puerta del MOSFET. En antiparalelo con cada resistencia de puerta (R_G) se ha insertado un diodo de pequeña señal modelo 1N4148, que hace que la descarga de puerta, y por tanto el apagado, sean lo más rápidos posibles.

En la *Figura 3.3* se muestra el driver de disparo junto con los elementos de protección comentados para el disparo de los MOSFET's.

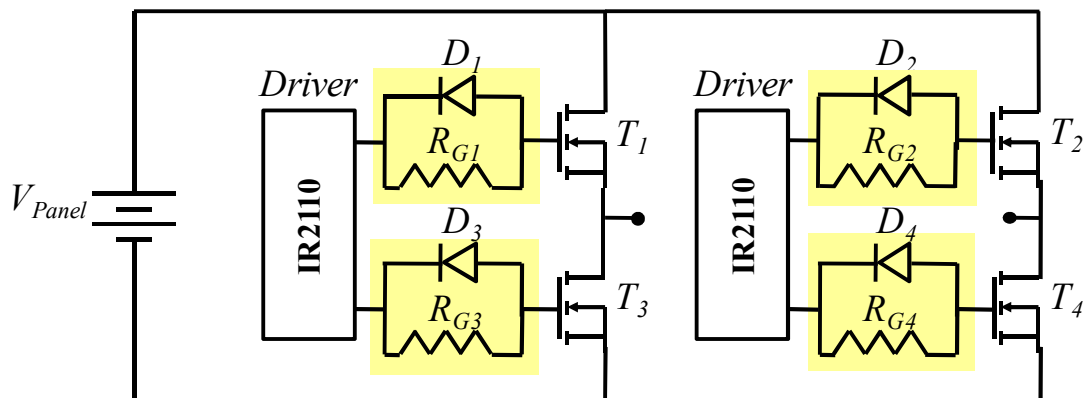


Figura 3.3. Driver de disparo y elementos de protección para el disparo de los MOSFET's

C. EL CONDENSADOR DE ENTRADA

A la entrada del inversor se ha añadido una batería de condensadores que hacen una capacidad total de $440 \mu F$, soportando hasta $450 V_{DC}$. Este valor de tensión máxima se ha escogido teniendo en cuenta que según las especificaciones de diseño, no se superan los $175 V_{DC}$ de entrada en el inversor monofásico.



Las funciones más destacadas de esta batería de condensadores son:

- Filtrar la corriente de alta frecuencia que produce el puente inversor, de manera que no llegue al panel fotovoltaico y pueda dañarlo.
- Es el elemento más importante para el control del MPPT, cuyo funcionamiento se basa en los tiempos de descarga del condensador en función de la potencia entregada por el generador fotovoltaico en determinados instantes.

D. EL DISIPADOR

El circuito inversor es la parte del hardware en la que coexisten mayores caídas de tensiones y corrientes circulantes, lo que implica elevadas pérdidas en forma de calor. Por esta razón, se han tenido en cuenta en su diseño diversos métodos que ayudan a la disipación del calor como pistas suficientemente gruesas, capas de masa amplias y la adición de elementos disipadores.

En este aspecto, el elemento más importante introducido en el diseño es un disipador de aluminio, que permite la disipación de potencia en forma de calor que se produce en los transistores. Este disipador, que cumple a su vez la función de elemento de fijación mecánica entre el circuito inversor y el conjunto transformador-filtro (Ver *Figura 3.4*), se ha sobredimensionado. Esto es así, a pesar de que una de las especificaciones de diseño es que el tamaño sea lo más reducido posible, debido a que es el transformador quien limita el tamaño mínimo del conjunto. En este caso se ha empleado un disipador que cubre completamente la cara más estrecha del transformado. De esta forma, se asegura una correcta disipación térmica de los MOSFET's, cuyo requerimiento máximo suponiendo un rendimiento medio del inversor del 80%, exigiría pérdidas térmicas máximas de 200 W, correspondiente a la máxima potencia (1 kW).

Además de evacuar calor de los transistores, el disipador es el elemento de unión entre la PCB donde está situado el puente inversor y el transformador de aislamiento, convirtiéndose en un soporte suficientemente rígido para la sustentación de la etapa inversora.

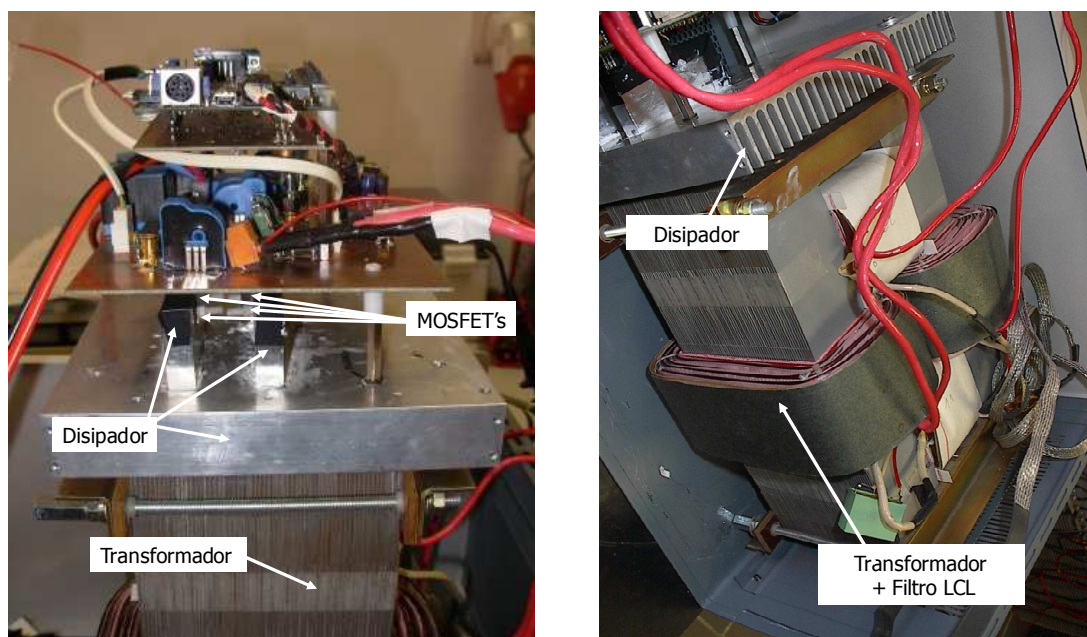


Figura 3.4. Disipador empleado para las pérdidas de los MOSFET's.

E. LA PCB DE LA ETAPA DE POTENCIA

En este proyecto se ha realizado el diseño de un circuito impreso que incluye toda la circuitería y dispositivos correspondientes a la etapa de potencia junto con el bloque de componentes correspondiente al sensado de los parámetros eléctricos.

La principal consideración que se ha tenido en cuenta para la realización de este diseño ha sido el emplear 2 planos de masa bien diferenciados: Uno para la etapa de potencia y todos aquellos dispositivos por los que circulen corrientes elevadas; y el otro para la parte de señal, por donde circulan pequeñas corrientes. Para poder llevar a cabo esta especificación, los componentes se han recolocado de tal manera que quede una separación bien definida entre ambas masas, que se unen en un único punto. De esta forma, se evita que circulen corrientes elevadas procedente de la etapa de potencia por la parte de señal.

En la *Figura 3.5* se muestra una imagen de la PCB de la etapa de potencia con la colocación de los principales componentes.

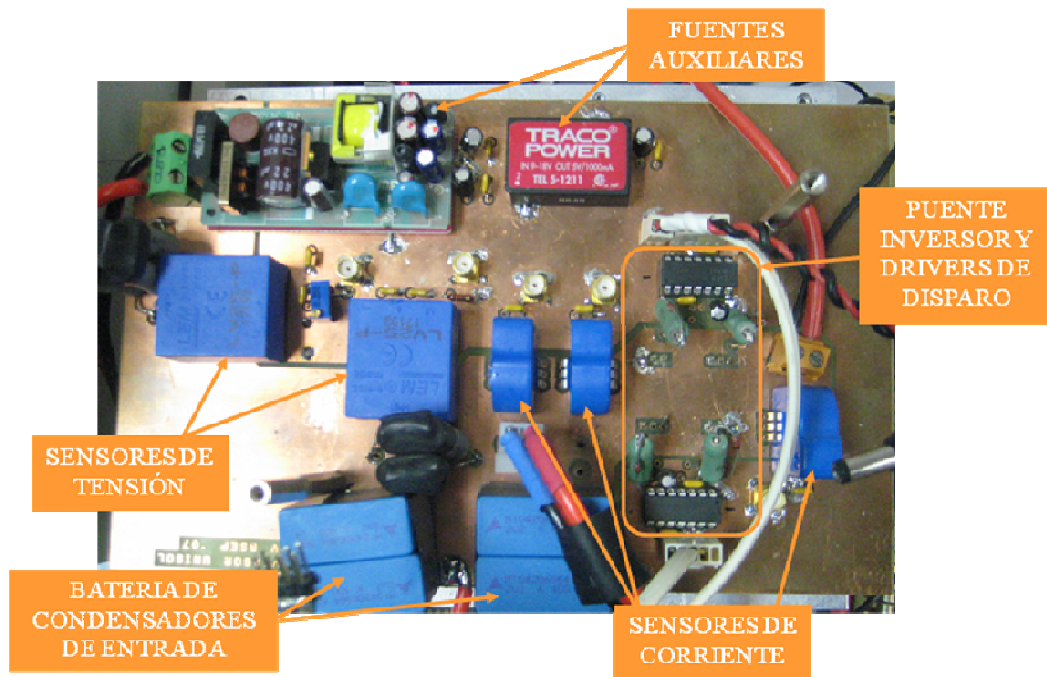


Figura 3.5. PCB con la etapa de potencia y el bloque de sensado del prototipo de pruebas del inversor monofásico.

Por otra parte, en la Figura 3.6 se representa el layout del plano superior de la placa de potencia. En esta imagen se puede observar la separación de los 2 planos de masa correspondientes a las zonas con mayores y menores corrientes circulantes, así como el punto de unión entre ambas.

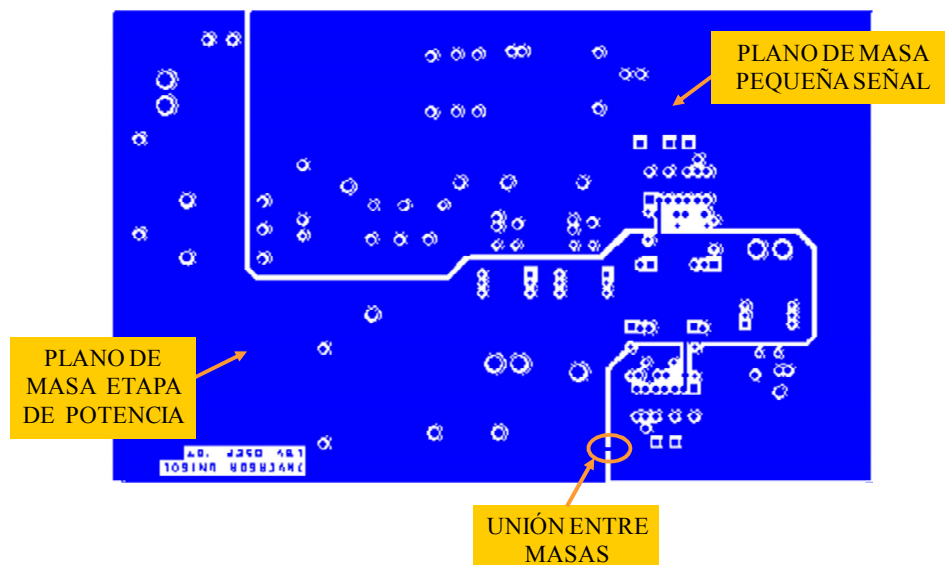


Figura 3.6. Layout de la PCB con los distintos planos de masa empleados

3.3. DISEÑO DEL SISTEMA DE MEDICIÓN Y CONTROL

El inversor del proyecto UNISOL lleva asociado un bloque de control digital basado en FPGA. La implementación de este sistema requiere para su funcionamiento el empleo de un circuito de sensado que es el encargado de medir las principales magnitudes eléctricas del sistema. A su vez, se ha diseñado un circuito de conversión que convierte las señales analógicas medidas por los sensores a valores digitales que sean procesables por el circuito de control. El control propiamente dicho es llevado a cabo mediante una FPGA, que genera las 4 señales de disparo para los transistores del inversor.

En la *Figura 3.7* se representa un diagrama de bloques con los principales elementos del sistema de control del inversor.

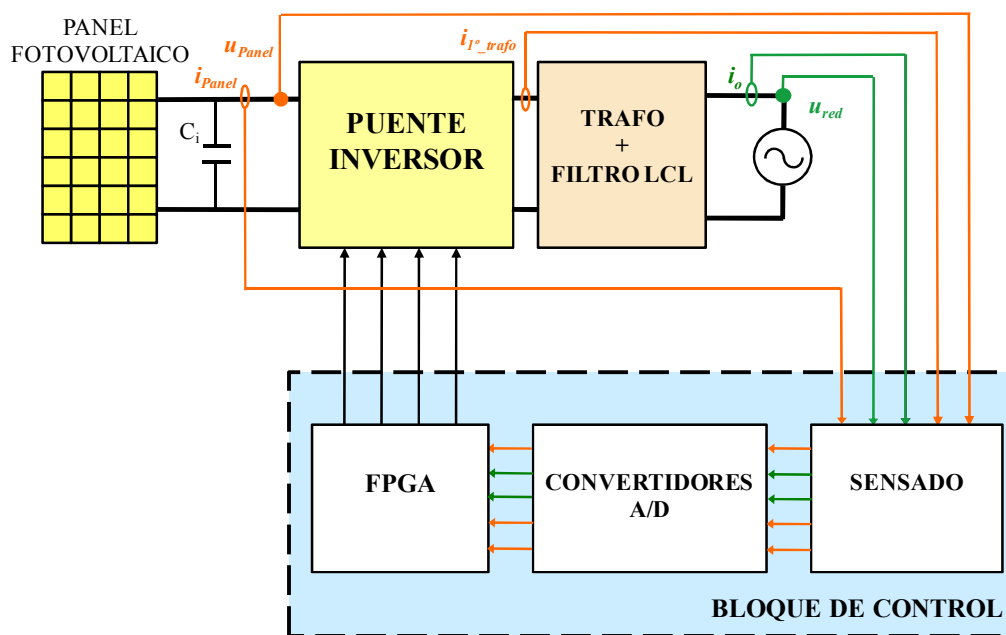


Figura 3.7. Diagrama de bloques del sistema de control.

3.3.1. EL BLOQUE DE SENSADO Y ACONDICIONAMIENTO.

El bloque de sensado está constituido por una serie de sensores de tensión y corriente que tienen la función de captar los parámetros eléctricos más importantes del sistema.

En el sistema de control se han añadido sensores para la medición de 5 parámetros eléctricos (ver *Figura 3.7*), que son:

- Tensión de red, u_{red} .
- Corriente de salida, i_o .
- Corriente del panel fotovoltaico, i_{panel} .
- Tensión del panel fotovoltaico, u_{panel} .
- Corriente del primario del trafo, $i_{1^\circ trafo}$.

De estos sensores, el prototipo construido solo emplea dos de ellos para su funcionamiento en lazo cerrado. Los sensores utilizados son los que miden la tensión de red y la corriente de salida. El resto no se emplean en las pruebas experimentales del prototipo.

Cada sensor de medida lleva asociado un circuito de acondicionamiento de señal que se encarga de tratar la señal analógica obtenida del sensor, imponiendo que la salida del mismo esté limitada en un rango de entre 0 y 2.5 V. Esta limitación viene impuesta por el convertidor analógico-digital empleado, que solo puede convertir señales que se encuentren entre estos límites.

El esquemático del circuito de sensado se encuentra incluido en el de la etapa de potencia. Ver *ANEXO I*.

3.3.1.1. SENSADO DE TENSIÓN.

El sistema de control del inversor requiere, para su funcionamiento en lazo cerrado, la medición de una serie de parámetros eléctricos entre los cuales se encuentra la tensión de la red (u_{red}).

El elemento empleado para medir tensiones es un transductor de tensión modelo LV 25-P (Ver *Figura 3.8*), basado en un sensor de corriente de efecto Hall. Este dispositivo dispone de aislamiento galvánico y es capaz de medir tensiones nominales de entre 10 y 500V.



Figura 3.8. Dispositivo transductor de tensión LV 25-P.

El principio de funcionamiento de este sensor es el siguiente: Se comporta como un transformador que genera una “corriente del lado secundario” proporcional a la del “lado primario”.

Este elemento de medida lleva asociado el circuito de conexión que se muestra en la Figura 3.9, que consiste en dos elementos resistivos situados a la entrada y a la salida del sensor. La resistencia R_1 tiene como misión transformar la tensión que se quiere medir en una corriente ($i_{primario}$), que es lo que el sensor es capaz de medir realmente. La resistencia de medida R_M , situada a la salida del sensor, permite adaptar la corriente de salida que proporciona el sensor, a los valores de tensión de salida según los requerimientos del diseño.

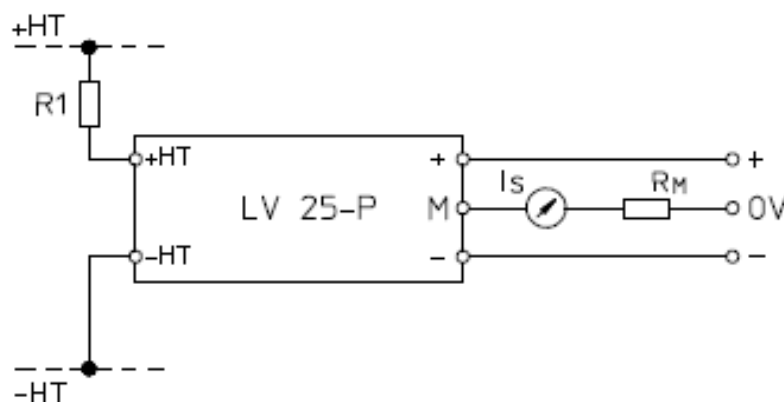


Figura 3.9. Circuito de conexión del sensor de efecto Hall transductor de tensión.

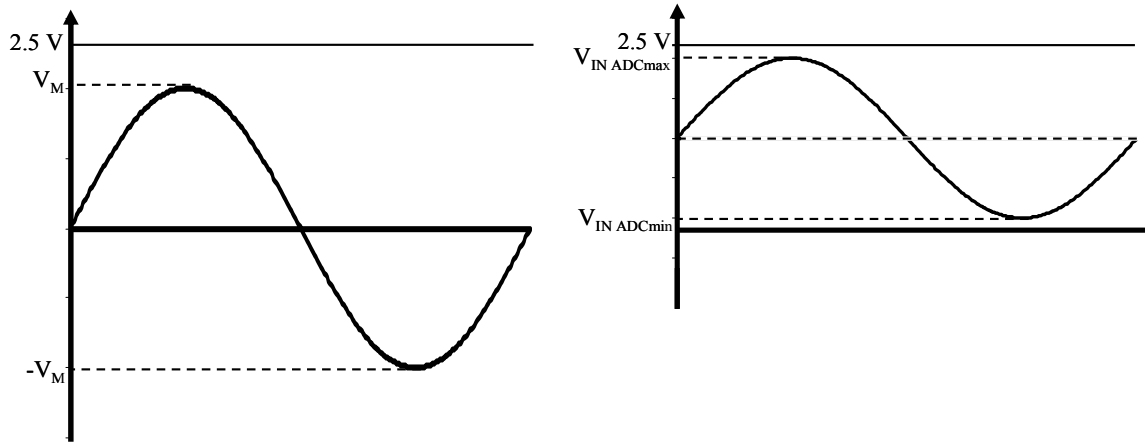


En el caso de la medida de tensión de red, se considera para el sensado una red eléctrica con las características propias de la red europea, es decir, 220 V eficaces a una frecuencia de 50 Hz. De este modo, el valor máximo que ha de ser capaz de medir el sensor es de 311 V de pico. Para dicho valor, y atendiendo a las hojas de características suministradas por el fabricante, la corriente por el primario debe ser de aproximadamente 10mA. El valor de la resistencia R_1 para estas condiciones es de 31.1 k Ω . Además, hay que tener en cuenta para la elección de este elemento, que la potencia máxima disipada será el producto de la tensión eficaz de red por la corriente que circula por esa resistencia, es decir, $P_{R1} = 2.2$ W.

Para el cálculo de R_M hay que tener en cuenta que, según las hojas de catálogo, ha de estar entre 100 y 350 Ω para la tensión de alimentación de ± 15 V, que es la empleada en este caso. Teniendo en cuenta que la corriente de salida máxima del transductor, I_{Smax} , es de ± 10 mA, se ha incluido un potenciómetro que permita ajustar el valor de la resistencia para que los valores de salida del sensor se encuentren dentro de unos límites aceptables por el convertidor A/D (Ver *Figura 3.10*).

CIRCUITO DE ACONDICIONAMIENTO DE LA TENSIÓN SENSADA

Dado que el control del inversor diseñado en el proyecto UNISOL es digital, las señales analógicas de los sensores, ya sean de tensión o de corriente, tienen que ser convertidas a valores digitales procesables por la FPGA. Para realizar esta transformación se han empleado convertidores A/D serie de 10 bits de entrada unipolar y cuyo valor máximo de entrada se ha limitado por diseño a 2.5 V. Sin embargo, las tensiones a la salida de los sensores son bipolares, es decir, incluyen valores negativos. Por tanto, precisan de un circuito de acondicionamiento que adapte esta señal al rango permitido por los ADC's (Ver *Figura 3.10*).



a. Tensión de red sensada con el transductor LV 25-P.

b. Tensión de red sensada después del circuito de acondicionamiento o tensión de entrada del ADC.

Figura 3.10. Tensión de red sensada antes y después del circuito de acondicionamiento

El circuito de acondicionamiento de los sensores de tensión se puede observar en la Figura 3.11. Este circuito consiste básicamente en la introducción de un diodo Zener, una resistencia de polarización para dicho Zener, R_A y un potenciómetro R_{POT} que, junto con la resistencia de medida, R_M , permiten introducir un offset y ajustar la escala a los límites deseados.

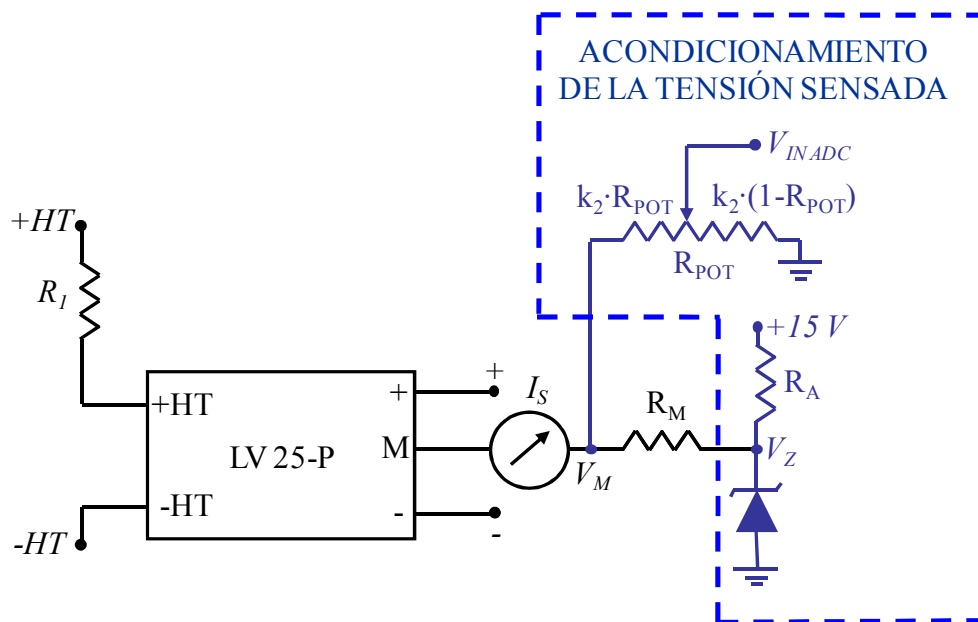


Figura 3.11. Circuito de acondicionamiento del sensor de tensión.

La ecuación de partida para el cálculo de R_M y R_{POT} es la *Ecuación 3.1*, que relaciona la tensión de entrada del convertidor A/D (V_{IN_ADC}) con la tensión vista desde la entrada del potenciómetro (V_{Th}).

$$V_{IN_ADC} = V_{Th} \cdot \frac{(1 - k_2) \cdot R_{POT}}{R_{POT} + R_{Th}} \quad \text{Ecuación 3.1}$$

En la *Figura 3.12* se representa el circuito equivalente de Thevenin visto desde la entrada del potenciómetro.

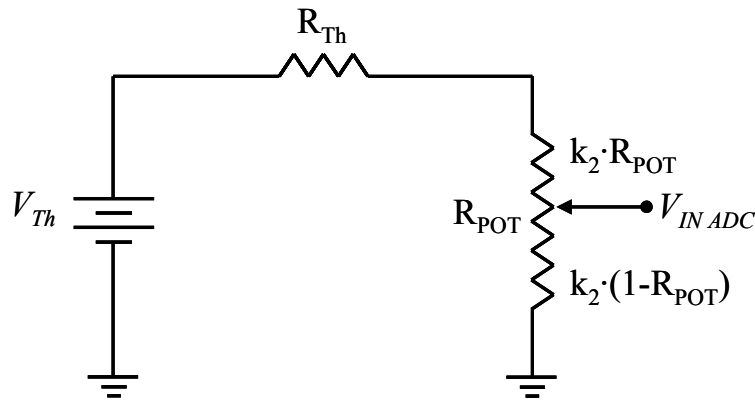


Figura 3.12. Equivalente de Thevenin para el circuito de acondicionamiento de la tensión sensada.

Los valores de V_{Th} y R_{Th} obtenidos a partir del equivalente de Thevenin son:

$$R_{Th} = R_M \quad V_{Th} = V_Z + I_S \cdot R_M$$

Sustituyendo estos valores de V_{Th} y R_{Th} en la, se obtiene la *Ecuación 3.2*:

$$V_{IN_ADC} = (V_Z + I_S \cdot R_M) \cdot \frac{(1 - k_2) \cdot R_{POT}}{R_{POT} + R_M} \quad \text{Ecuación 3.2}$$

Partiendo de esta ecuación, se pueden obtener las expresiones de la ganancia (*Ecuación 3.3*) y el offset (*Ecuación 3.4*), que son los parámetros necesarios para obtener la señal adecuada a la entrada del ADC.

$$\text{Ganancia} = \frac{\partial V_{\text{IN_ADC}}}{\partial I_s} = (1 - k_2) \cdot \frac{R_{\text{POT}}}{R_{\text{POT}} + R_M} \cdot R_M \quad \text{Ecuación 3.3}$$

$$\text{Offset} = V_{\text{IN_ADC}}|_{I_s=0} = V_Z \cdot (1 - k_2) \cdot \frac{R_{\text{POT}}}{R_{\text{POT}} + R_M} \quad \text{Ecuación 3.4}$$

Los cálculos referentes al circuito de acondicionamiento de la tensión de red sensada se encuentran en el *ANEXO III*.

En la *Figura 3.10* se puede observar el efecto que tiene introducir el circuito de acondicionamiento a la salida del sensor de u_{RED} . En la *Figura 3.10.a* se representa la tensión de entrada del ADC sin circuito de acondicionamiento, mientras que en la *Figura 3.10.b* se puede apreciar el offset y el cambio de escala producidos en la señal de tensión de red sensada al añadir el circuito de acondicionamiento.

3.3.1.2. SENSADO DE CORRIENTE

El sistema de control diseñado dispone de 3 canales para la medición de corriente. Sin embargo, en este documento solo se describe uno de ellos, que es el referente a la corriente entregada por la red (i_o), por ser el más importante. El sensado de este parámetro se realiza con un dispositivo transductor de corriente de efecto Hall modelo LTS 25-NP (ver *Figura 3.13*), cuyo comportamiento lineal característico se representa en la *Figura 3.14*.

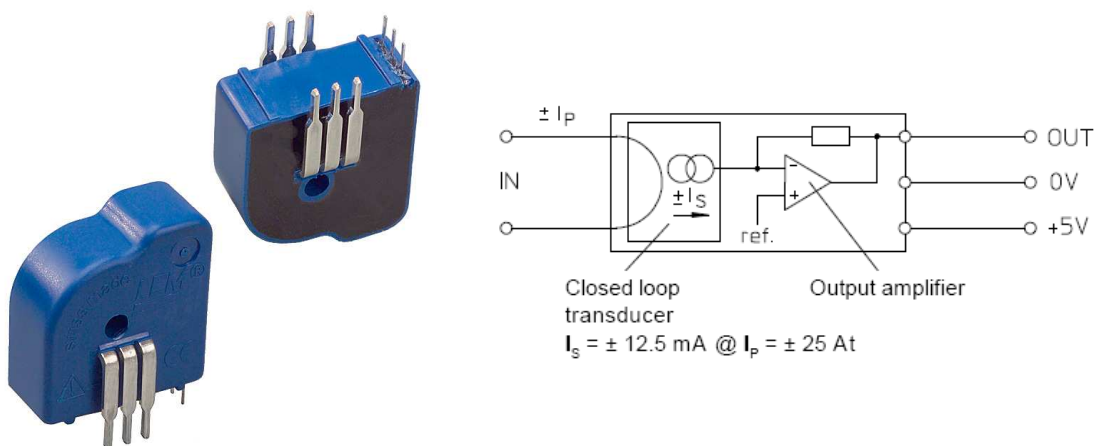


Figura 3.13. Dispositivo transductor de corriente LTS 25-NP

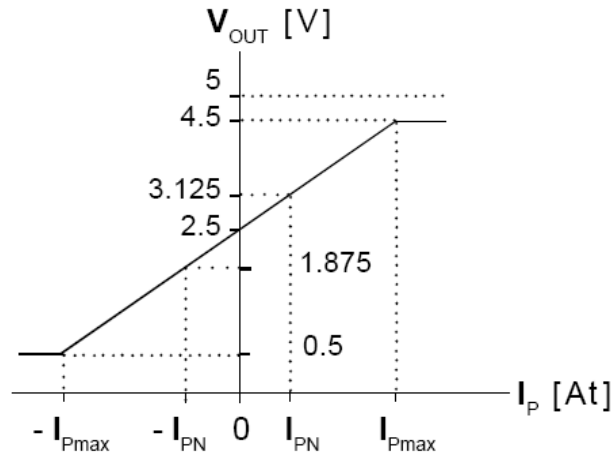


Figura 3.14. Característica de funcionamiento del sensor de corriente. Curva de tensión de salida (V_{OUT}) frente a corriente medida (I_P)

El LTS 25-NP permite varias configuraciones de sus terminales de entrada en función de la corriente que se desea medir. La configuración inicial de los sensores escogida ha sido la correspondiente al menor nivel de corriente, de 8 A (permite las configuraciones de 12 A y 25 A). De esta manera, se pueden hacer pruebas con el prototipo para su puesta en marcha y depuración con niveles de corriente bajos, aprovechando mejor la resolución de los ADC. Nótese que si se utiliza un rango de corrientes muy elevado en el sensor, la resolución de la medida es menor (menos amperios por bit) y por lo tanto las pruebas con corrientes pequeñas son más difíciles.

Independientemente del rango de corriente escogido, la tensión de salida del sensor es de 2.5 ± 0.600 V (según Figura 3.14, obtenida de las hojas de catálogo), es decir:

$$V_{0_SENSORmax} = 3.2 \text{ V}$$

$$V_{0_SENSORmin} = 1.9 \text{ V}$$

Sin embargo, estas tensiones no son válidas para la conversión de datos de los ADC's, luego se requiere un bloque de acondicionamiento de señal para convertirlas en valores que se encuentren dentro de los límites aceptables por los ADC's.

CIRCUITO DE ACONDICIONAMIENTO DE LA CORRIENTE SENSADA

Al igual que el sensor de tensión, el sensor de corriente precisa de un circuito de acondicionamiento que añada un determinado offset a la señal de salida del sensor para que sus límites se encuentren dentro del rango de conversión impuesto al convertidor A/D, es decir, en valores entre 0 y 2.5 V.

El circuito de acondicionamiento de señal para los sensores de corriente se ha integrado con un buffer de entrada que forma parte del circuito de acondicionamiento de señal propio de los convertidores A/D (*Figura 3.15*). Este circuito se encuentra situado justo a la salida del sensor de corriente.

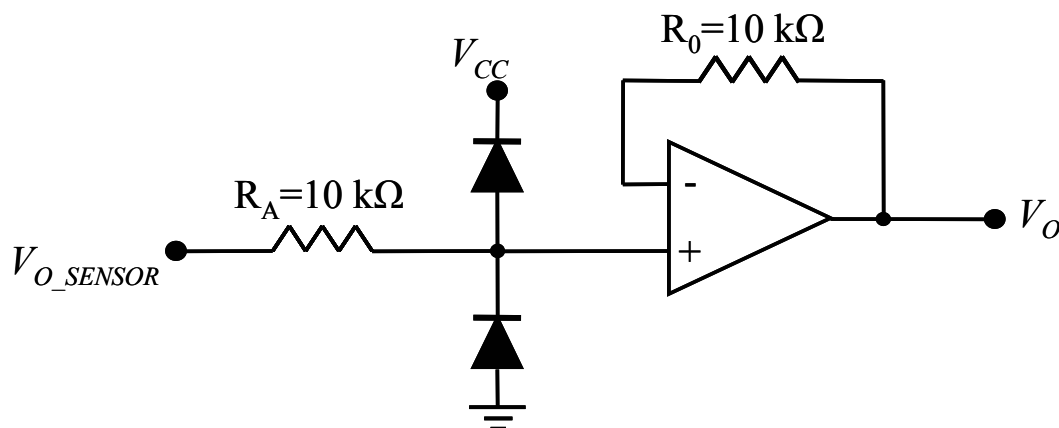


Figura 3.15. Buffer de entrada del circuito convertidor A/D.

El circuito de cambio de escala y adición de offset de la señal de salida del sensor se muestra en la *Figura 3.16*. Las resistencias determinan el escalado y offset que se le aplica a la señal del sensor, mientras que los diodos protegen al circuito de tensiones demasiado grandes o negativas.

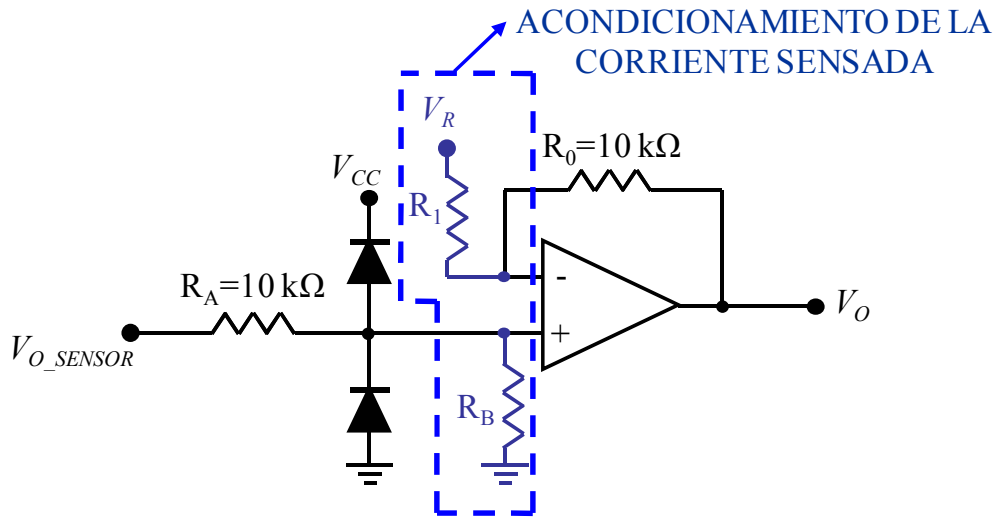


Figura 3.16. Modificación del buffer de entrada para el circuito de acondicionamiento de la corriente sensada.

La obtención de los valores adecuados de las resistencias R_1 y R_B se aborda a partir de la *Ecuación 3.5*, que representa la tensión de salida (V_O) del buffer de entrada:

$$V_O = V_+ \cdot \left(1 + \frac{R_O}{R_1}\right) - V_R \cdot \frac{R_O}{R_1} \quad \text{Ecuación 3.5}$$

Donde:

$$V_+ = V_{\text{Osensor}} \cdot \frac{R_B}{R_A + R_B}$$

De este modo, se tiene que:

$$V_O = V_{\text{Osensor}} \cdot \left(\frac{R_B}{R_A + R_B}\right) \cdot \left(1 + \frac{R_O}{R_1}\right) - V_R \cdot \frac{R_O}{R_1} \quad \text{Ecuación 3.6}$$

En la *Ecuación 3.6* son conocidos R_O y R_A ; y como tensión V_R se ha empleado la propia alimentación de la fuente, de valor 3.3 V.

Los cálculos realizados para obtener los valores de los diversos componentes que forman parte del circuito de acondicionamiento de la corriente de salida sensada se encuentran en el *ANEXO IV*.

El resultado final de los cálculos es el circuito de acondicionamiento representado en la *Figura 3.17*, donde el buffer de entrada del convertidor A/D ha quedado convertido en un circuito amplificador con un offset.

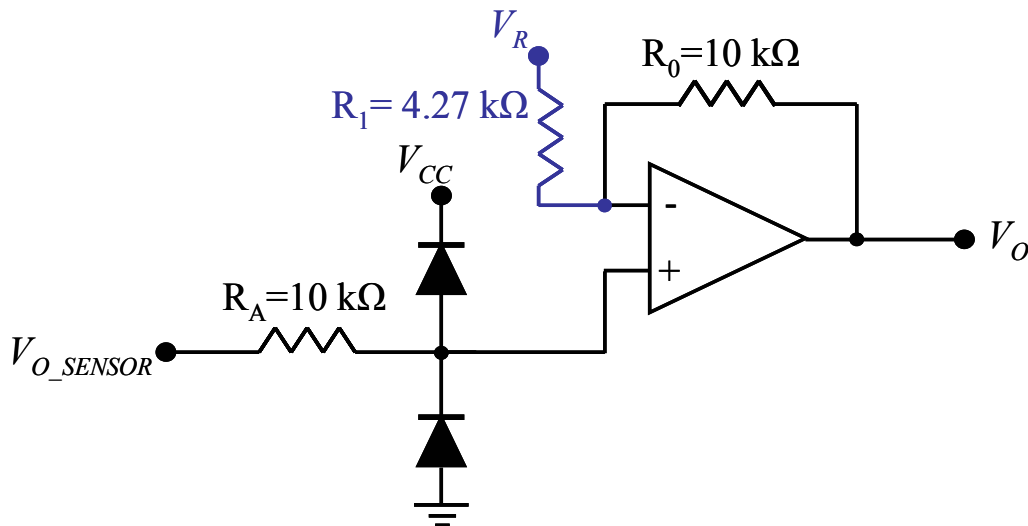
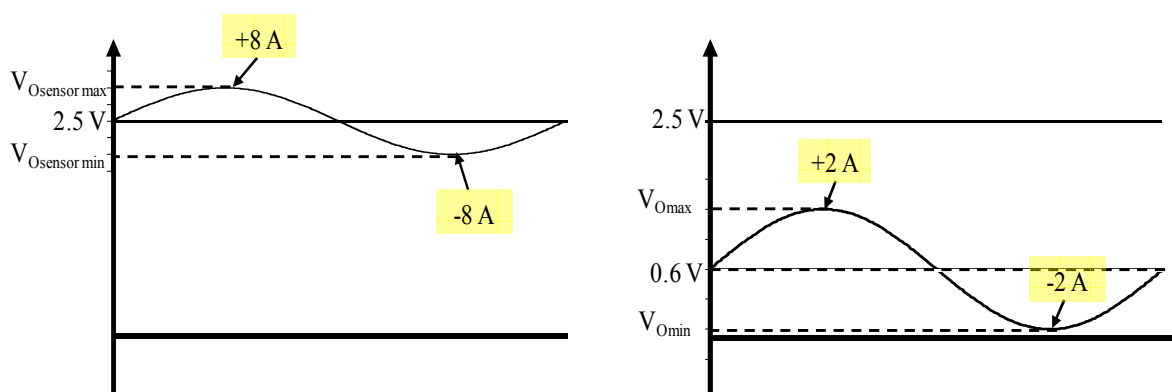


Figura 3.17. Circuito de acondicionamiento de la corriente i_o sensada.

En la *Figura 3.18* se puede apreciar cómo, al introducir el circuito de acondicionamiento a la salida del sensor de corriente, la señal que originariamente estaba centrada en 2.5 V pasa a estarlo en 0.6 V, experimentando además un cambio de escala.



a. Corriente de red sensada con el transductor LTS 25-NP.

b. Corriente de red sensada después del circuito de acondicionamiento.

Figura 3.18. Corriente entregada a la red, sensada antes y después del circuito de acondicionamiento.

3.3.2. EL CIRCUITO DE CONVERSIÓN DE DATOS

El bloque de conversión de datos se encarga de digitalizar las señales analógicas una vez que han sido sensadas y acondicionadas, para posteriormente enviar estos datos a la FPGA. El circuito de conversión de datos mostrado en la *Figura 3.19* se ha realizado a partir del esquemático del módulo periférico PmodAD1, del fabricante DIGILENT, basado en un convertidor A/D serie comercial modelo ADS7887. La circuitería y el layout se han adaptado a las necesidades del diseño (ver *ANEXO II*). En este esquemático se han incluido un total de 5 canales, de manera que se puedan medir el mayor número de parámetros eléctricos posibles en caso de ser necesario.

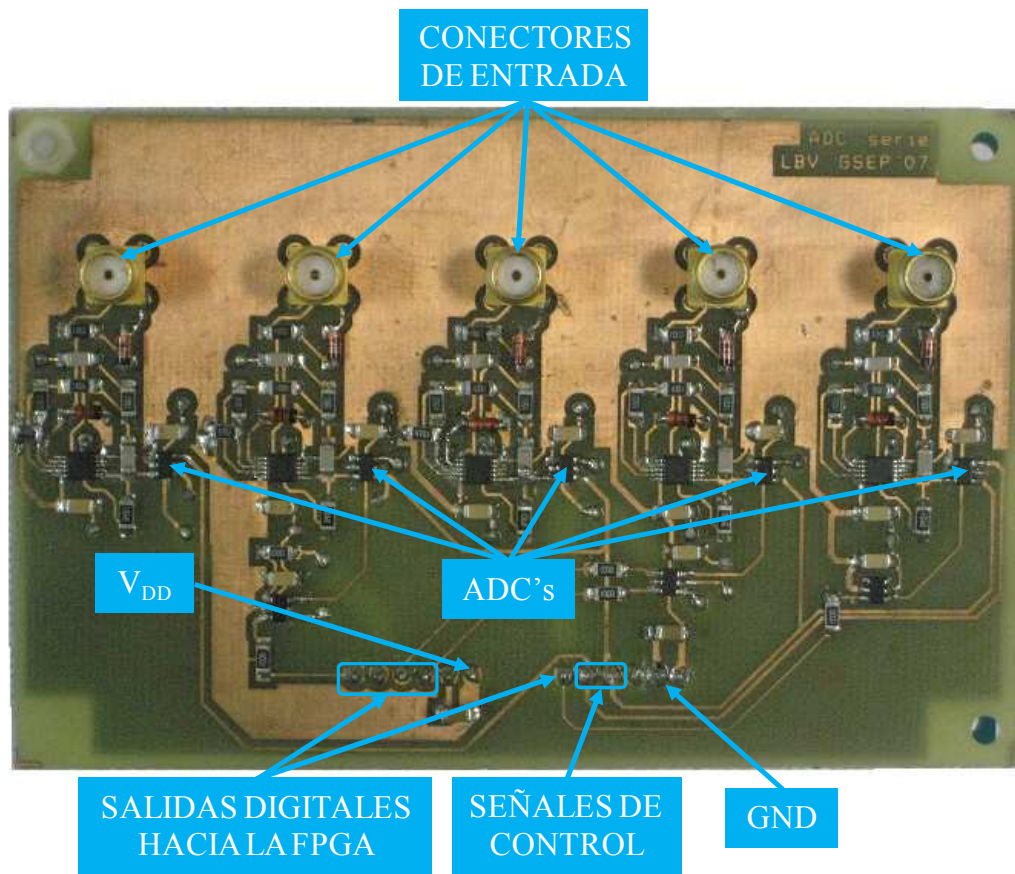
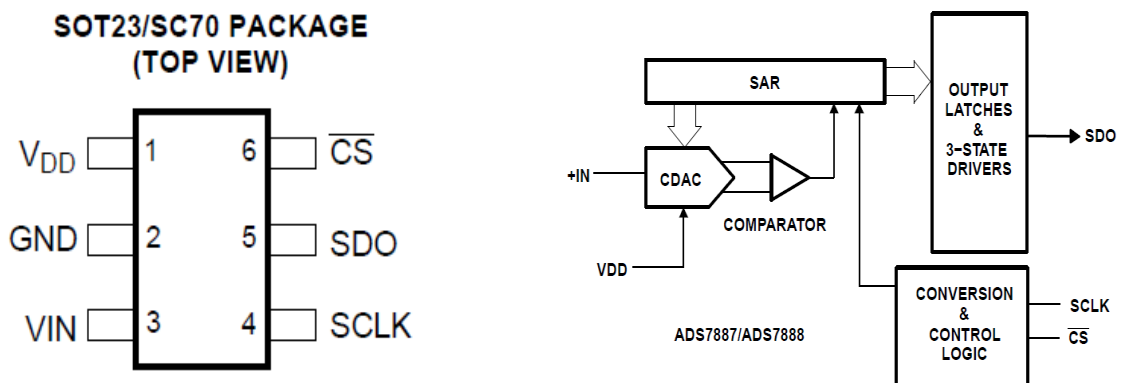


Figura 3.19. Circuito de conversión de señales analógicas a digitales.

El dispositivo principal del circuito de conversión es el ADS7887. Las especificaciones principales de este convertidor A/D son las siguientes:

- Interfaz serie (hasta 25 MHz)
- 10 bits de resolución.
- 1.25 MHz de frecuencia de muestreo máxima (1.25 millones de muestras por segundo).
- Tensiones de entrada unipolares (de 0 V a la tensión de alimentación, siendo $V_{DD}=5.25$ V).

El convertidor ADS7887 se muestra en la *Figura 3.20*. Este elemento dispone de una entrada analógica (VIN), por la que se recibe la señal analógica unipolar a convertir. Este dispositivo es controlado a través de la FPGA, que se encarga de mandar la señal de muestreo, conocida como Chip Select (\overline{CS}) y de enviar los flancos de reloj (SCLK) que sincronizan el envío de cada bit. La cadena de bits correspondiente a cada conversión realizada es enviada hacia la FPGA mediante la salida digital SDO.



a. Patillaje del convertidor ADS7887.

b. Esquema de bloques interno del ADS7887.

Figura 3.20. El convertidor A/D modelo ADS7887.

El cronograma de funcionamiento del ADC modelo ADS7887 se muestra en la *Figura 3.21*. Tras recibirse un flanco de bajada de \overline{CS} , la señal de entrada es muestreada y comienza el proceso de conversión. Mientras la conversión está en proceso, la salida digital SDO envía una cadena de datos que está constituida por: 4 ceros de cabecera, seguidos de la palabra digital de 10 bits y 2 ceros de cola.

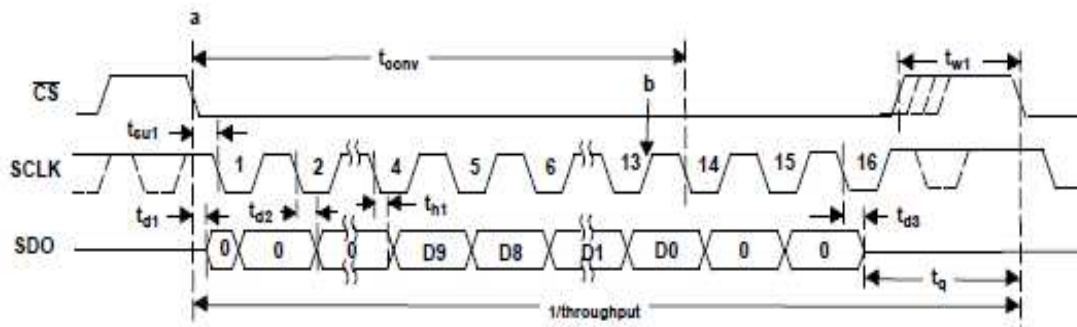


Figura 3.21. Cronograma de tiempos para el funcionamiento del convertidor ADS7887.

Cada canal de conversión A/D dispone de un buffer de entrada seguido de un filtro antialiasing que elimina aquellas frecuencias que superan la frecuencia crítica, siendo ésta la mitad de la frecuencia de muestreo. El objetivo principal de este tipo de filtro es el de poder reconstruir correctamente cualquier señal muestreada por el ADC, evitando las distorsiones que se pudieran producir por el efecto de aliasing.

En la Figura 3.22 se representa el esquema eléctrico del convertidor A/D junto con el filtro antialiasing correspondiente y un buffer de ganancia unidad (Báscula de Schmitt) situado a la salida del ADC, que proporciona una baja impedancia a la salida del mismo.

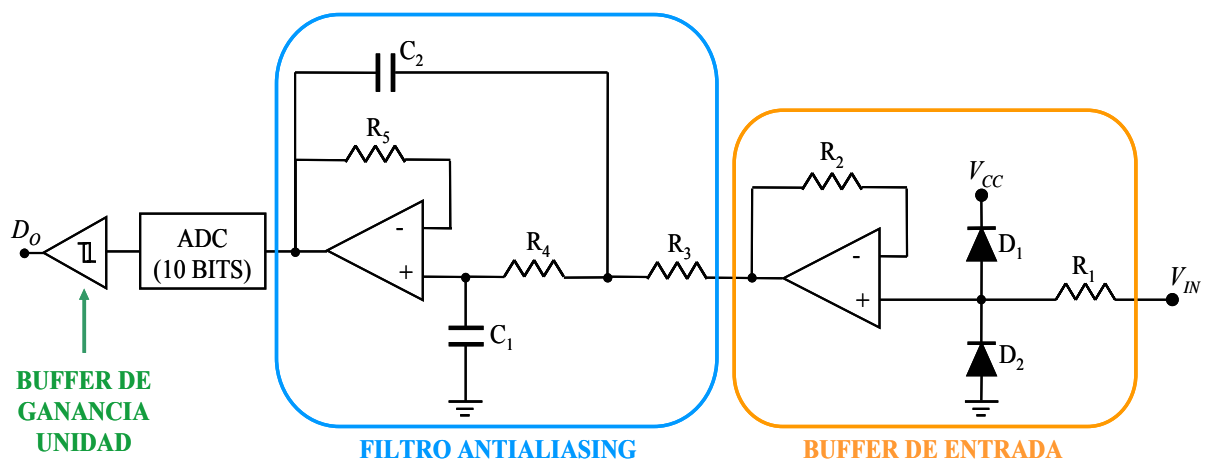


Figura 3.22. Esquema eléctrico del convertidor A/D.



3.3.3. LA FPGA

La FPGA es el dispositivo encargado de gestionar el sistema de control del inversor. En este proyecto se ha empleado como sistema de control una FPGA incluida en una tarjeta de evaluación modelo BASYS, del fabricante DIGILENT. Esta tarjeta programable se caracteriza por ser la más baja de la gama de DIGILENT e incluye en su esquemático una FPGA modelo *Spartan 3E*, de XILINX, con un total de 100000 puertas lógicas. La programación de la misma se realiza a través del interfaz JTAG, empleando para ello un puerto USB2 incluido en la propia tarjeta de evaluación. De este modo, se facilita proceso de reprogramación del dispositivo.

En este proyecto, la FPGA tiene las misiones principales de controlar los ADC's y generar los disparos de los transistores trabajando en lazo cerrado.

A continuación se describen brevemente los principales elementos utilizados de la tarjeta de evaluación de la FPGA para la realización del proyecto UNISOL (Ver *Figura 3.23*):

- Uno de los pulsadores de esta tarjeta se emplea como Reset común para todo el sistema de control.
- Los interruptores tienen 2 funciones definidas: Uno de ellos se emplea para la puesta en marcha manual de los convertidores A/D, mientras que el resto de interruptores se utilizan para la modificación de los parámetros de control del lazo de corriente del inversor.
- Los pines de salida del puerto VGA se emplean para la monitorización de los registros digitales tratados por la FPGA. Para ello, se hace uso de un convertidor D/A diseñado exclusivamente para este proyecto. Este convertidor se basa en la implementación mediante VHDL de un circuito digital capaz de generar una señal PWM cuyo ancho de pulso es proporcional al valor del registro de 10 bits que se requiere medir. Dicha PWM se muestra a través de uno de los pines de salida del puerto VGA para posteriormente pasarla por un filtro que permite la representación analógica del valor digital.

- Los pines de entrada-salida digitales (I/O) de esta tarjeta de evaluación de FPGA están subdivididos en 4 grupos, y se emplean fundamentalmente con 3 finalidades:
 - Uno de los grupos de 4 pines se emplea como salida para la generación del disparo de los transistores del puente inversor.
 - Los 2 grupos de pines de I/O centrales están orientados al control de los convertidores A/D. De estas señales, 5 de ellas son las entradas de la FPGA que reciben los paquetes de datos en serie de los convertidores A/D, mientras que otras 2 se emplean como salidas con las señales de control SCLK y CS. También se utilizan los pines de alimentación externa y masa de los que dispone esta tarjeta de evaluación para la alimentación de la tarjeta de los ADC's. El último grupo de pines de I/O disponibles en la FPGA tienen como función mostrar los 4 bits más significativos de cada conversión A/D realizada por uno de los convertidores. En concreto, se visualizan los bits más significativos correspondientes a la conversión de la tensión de red.

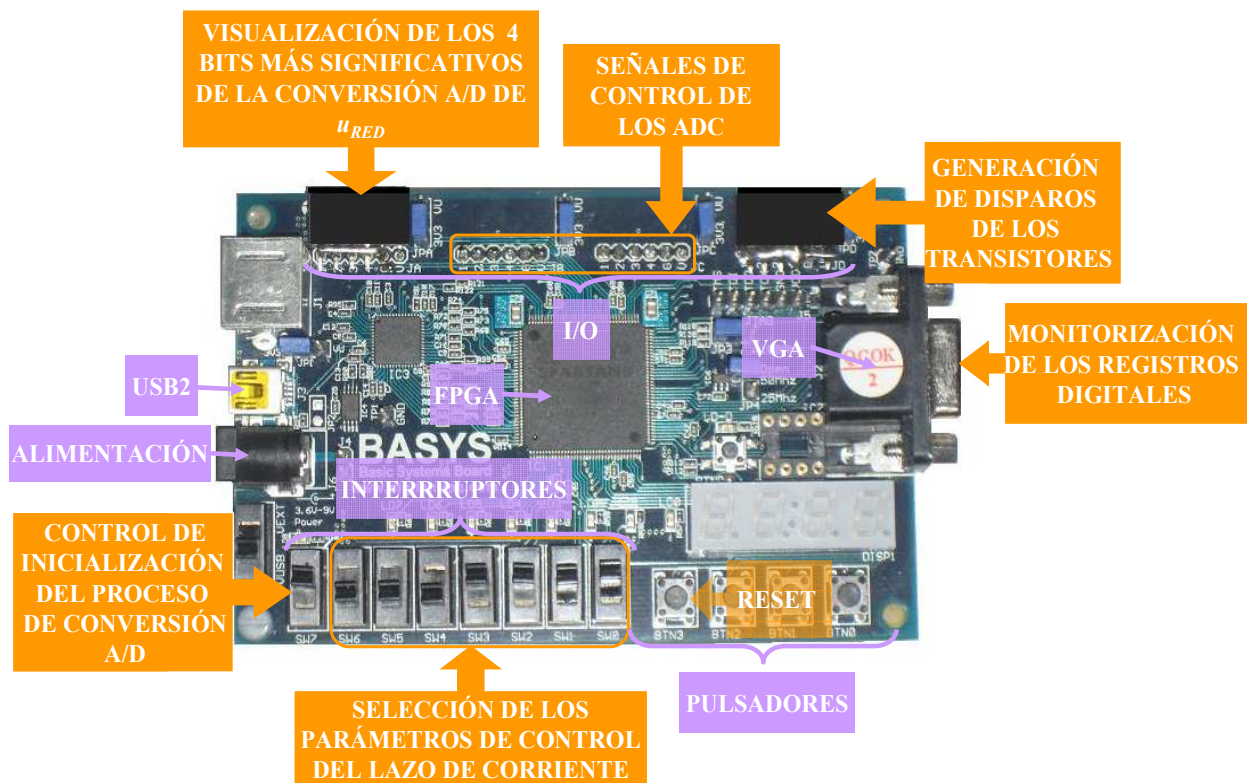


Figura 3.23. Tarjeta de evaluación FPGA empleada para el control del inversor y los ADC.

3.3.4. FUENTES AUXILIARES

Además de los principales dispositivos que forman parte del propio inversor y del control del mismo, es necesaria la incorporación en el diseño de las fuentes de alimentación requeridas por los siguientes dispositivos:

- **Los sensores de tensión:** Emplean una alimentación dual de entre ± 12 V y ± 15 V.
- **Los sensores de corriente:** Necesitan para su funcionamiento una tensión de 5V.
- **Los drivers para el disparo de los transistores:** Requieren dos alimentaciones diferentes, pudiendo emplear tensiones de hasta 25 V.
- **La FPGA:** Permite tensiones de entre 3.6 V y 9 V.
- **La tarjeta de los ADC's.** Este dispositivo está formado por diversos componentes electrónicos que comparten un rango común de tensiones de funcionamiento de entre 2.7 V y 5 V.

La selección de las fuentes de alimentación se ha realizado mediante un pequeño estudio de los consumos máximos de los dispositivos que se acaban de enumerar, que queda reflejado en la *Tabla 3.1*.

COMPONENTE	Corriente	Tensión	Cantidad	Potencia Estimada
LV 25-P	35 mA	± 15 V	2 ud	1,050 W
LTS 25-NP	24 mA	5 V	3 ud	0,370 W
IR2110	200 mA	5 V; 15 V	2 ud	6 W
Tarjeta FPGA	40 mA	5 V	1 ud	0,200 W
Placa ADC's	20 mA	3,3 V	1 ud	0,066 W
TOTAL				7,686 W

Tabla 3.1. Estimación del consumo de potencia máximo de los dispositivos del sistema de control.

A la vista de los resultados obtenidos en la *Tabla 3.1*, se aprecia una demanda de potencia máxima de casi 8 W para el conjunto completo de dispositivos que constituyen el sistema de control del inversor. Por otra parte, se han escogido tensiones estándar para la alimentación de cada dispositivo, pudiéndose observar 3 tipos de alimentación diferentes: una de ± 15 V, otra de 5 V y una última de 3.3 V para la placa de los ADC's.

La razón de emplear una tensión diferente de 5V en la tarjeta de conversión de los ADC's es que al estar interconectada con la FPGA, obtiene la alimentación directamente de los terminales de salida de la misma, que han sido configurados para ofrecer 3.3 V.

La conclusión del pequeño análisis realizado es que se necesita la incorporación en el diseño de 2 fuentes de ± 15 V y +5 V respectivamente, de manera que sean capaces de ofrecer entre ambas potencias de hasta 8W. Concretamente, se han seleccionado los modelos TOM 12215 y TEL 5-1211 de la marca TRACO (Ver *Figura 3.24*). La primera de ellas es un convertidor CA/CC que a partir de la tensión de red, de 220 V eficaces, impone a su salida ± 15 V. Este dispositivo es capaz de entregar 12 W, que es una potencia más que suficiente en comparación con los casi 8 W que se requieren. Por su parte, el TEL 5-1211 es un convertidor CC/CC que a partir de los 15 V de entrada obtenidos del convertidor CA/CC, impone a la salida +5 V, ofreciendo corrientes de hasta 1000 mA.



a. Convertidor CA/CC modelo TOM 12215



b. Convertidor CC/CC modelo TEL 5-1211

Figura 3.24. Fuentes de alimentación auxiliares.

3.4. MONTAJE DEL PROTOTIPO DE PRUEBAS

Tras haber definido los principales componentes del prototipo de pruebas diseñado en el proyecto UNISOL, el siguiente paso es el ensamblaje de todos ellos en un único bloque.

Para realizar el montaje del prototipo se han seguido los siguientes pasos:

- 1) La tarjeta de evaluación de la FPGA ha sufrido leves modificaciones que le permiten interactuar con la placa de los ADC's. En la *Figura 3.25* se puede apreciar la eliminación de los conectores centrales de la tarjeta de evaluación. Esta modificación se ha realizado para facilitar el ensamblaje directo entre la tarjeta de conversión de los ADC's y la FPGA.
- 2) Se ha incluido en la FPGA un conector montado sobre las tiras de pines de I/O laterales. Este conector tiene cortocircuitado su patillaje y a su vez está unido por soldadura a una de los pines de masa disponibles en la FPGA. El objetivo de esta modificación es hacer que las señales de disparo que van a los drivers tengan cada una su propio conductor de masa, quedando todos ellos unidos en un punto común de la FPGA.
- 3) Se han ensamblado la tarjeta de evaluación de la FPGA con la tarjeta de conversión de datos A/D a través de los pines de control de los ADC's (Ver *Figura 3.25*). Entre estos pines se encuentran aquellos que aportan la alimentación a la tarjeta de los ADC's. Además, se han añadido 2 separadores de plástico entre ambas tarjetas para conseguir una mayor sujeción mecánica entre ambas.

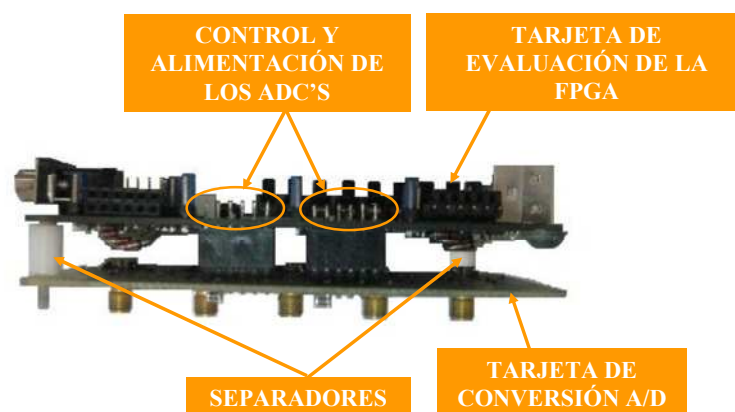


Figura 3.25. Conexión entre la tarjeta de evaluación de la FPGA y la tarjeta de conversión de los ADC's.

- 4) Se ha unido el conjunto de tarjeta de evaluación de la FPGA + tarjeta de conversión A/D a la placa de potencia mediante separadores metálicos y tornillos de plástico colocados en 2 esquinas opuestas de la PCB de potencia (ver *Figura 3.27*).



- 5) LA PCB principal, formada por la etapa de potencia y el bloque de sensado, se ensambla físicamente con el disipador de los MOSFET's. Para ello, se atornilla cada transistor a una pletina de aluminio que a su vez está atornillada al disipador aleteado. Además, esta PCB se encuentra anclada al disipador mediante tornillos y separadores metálicos por 2 puntos más, colocados en 2 extremos opuestos del propio disipador (ver *Figura 3.27*).
- 6) Se ha atornillado el elemento disipador a la bancada del conjunto Transformador + Filtro, empleando este último como elemento de sujeción del resto del sistema (ver *Figura 3.27*).
- 7) Una vez ensamblados los principales componentes, se han realizado las correspondientes conexiones entre los diferentes elementos del sistema. Para ello, se han realizado una serie de cables que se describen a continuación:
 - Cable de alimentación de la FPGA (ver *Figura 3.26.a*): Consiste en un cable rígido trenzado que sirve para alimentar a la tarjeta de evaluación de la FPGA a través de la PCB principal.
 - Cable para la transmisión de la señal de disparo entre la tarjeta de evaluación de la FPGA y la PCB de la etapa de potencia (Ver *Figura 3.26.b*): Este cable se ha construido a partir de 2 cables planos de 4 hilos. Ambos cables tienen una terminación común que va conectada a la tarjeta de la FPGA, donde se generan los disparos. Por su parte, el otro terminal de cada cable a su driver correspondiente (un IR2110 para cada rama), de manera que 2 de los hilos del cable se emplean para el disparo de cada transistor y los otros 2 están conectados a masa.
 - Cable de transmisión de señal entre el bloque de sensado y la tarjeta de conversión de los ADC's (ver *Figura 3.26.c*): Se trata de un cable apantallado cuyos extremos son conectores SMA.

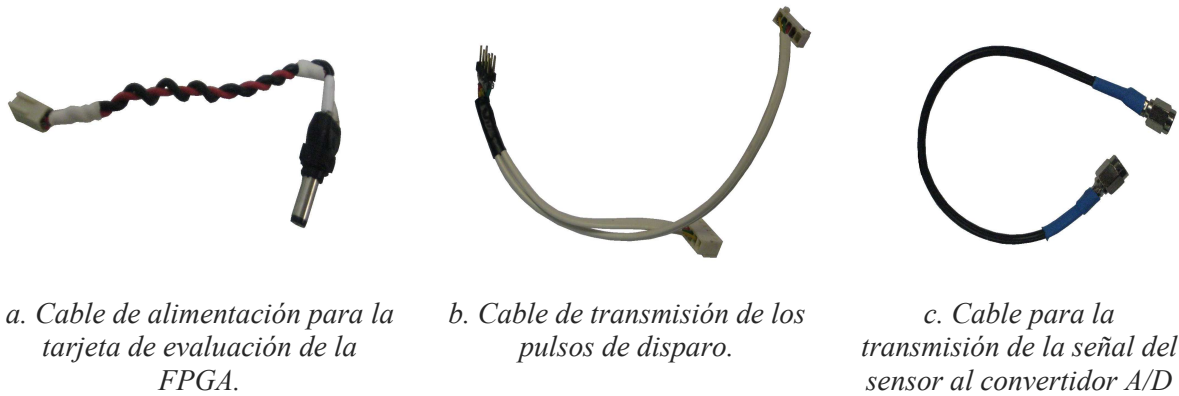


Figura 3.26. Cables de conexión entre los diferentes componentes del prototipo de pruebas.

En la Figura 3.27 se muestra el montaje completo del prototipo de pruebas diseñado para el proyecto UNISOL, donde se pueden observar los principales elementos descritos en este capítulo.

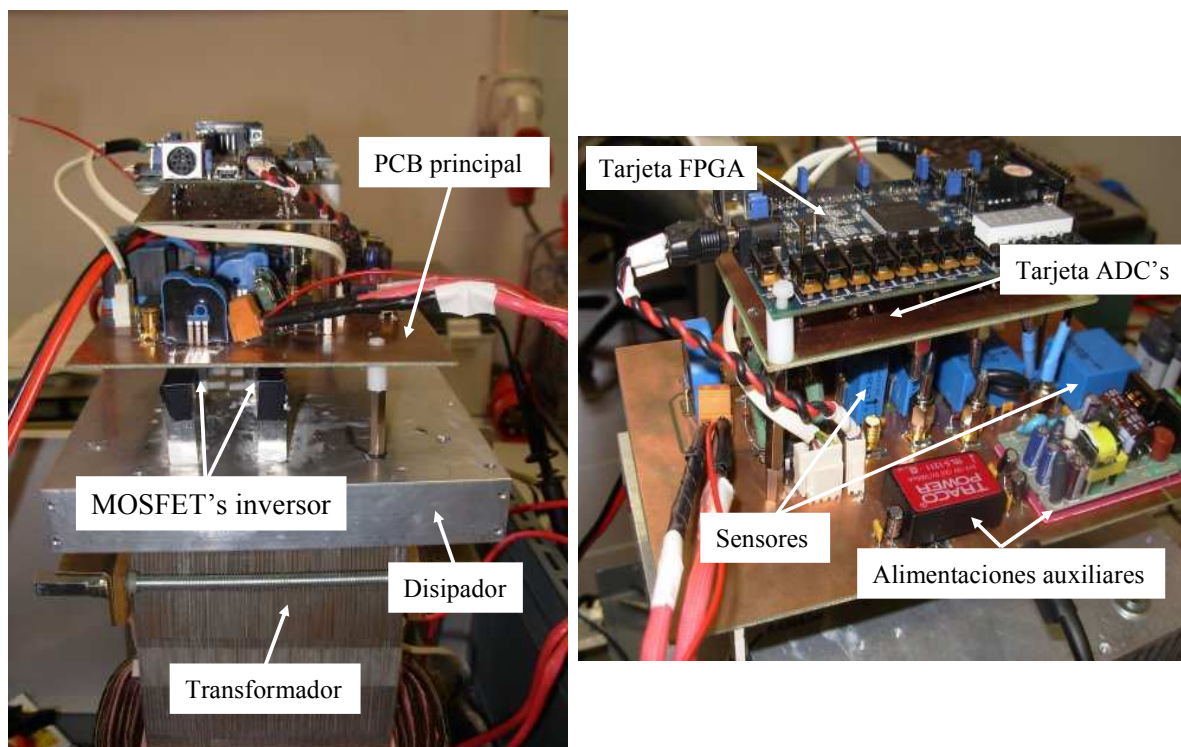


Figura 3.27. Montaje completo del prototipo de pruebas diseñado para el inversor monofásico del proyecto UNISOL.



CAPÍTULO 4.

CONTROL DIGITAL DEL INVERSOR EN LAZO ABIERTO Y EN LAZO CERRADO CONECTADO A RED

El inversor monofásico del proyecto UNISOL es controlado digitalmente mediante una FPGA, por lo que toda la documentación que se describe en este capítulo se centra en la implementación de los bloques digitales desarrollados mediante lenguaje VHDL. Ver *Figura 4.1*.

El diseño del hardware digital detallado en este capítulo se subdivide en varios apartados: En primer lugar se realiza una descripción del sistema de control para el funcionamiento del inversor en bucle abierto, tratando de este modo de validar experimentalmente el funcionamiento del hardware del prototipo completo. En segundo lugar, se hace una breve descripción acerca del hardware digital implementado para la gestión de la tarjeta de conversión de los ADC's, necesaria para cerrar el lazo. A continuación, se describe el control para el inversor trabajando en lazo cerrado, donde se incluye un lazo de corriente para la regulación de la corriente de salida.

Cada bloque de control digital se ha estudiado inicialmente mediante la herramienta de simulación MODELSIM y posteriormente se ha validado experimentalmente con el prototipo de pruebas diseñado en este proyecto.

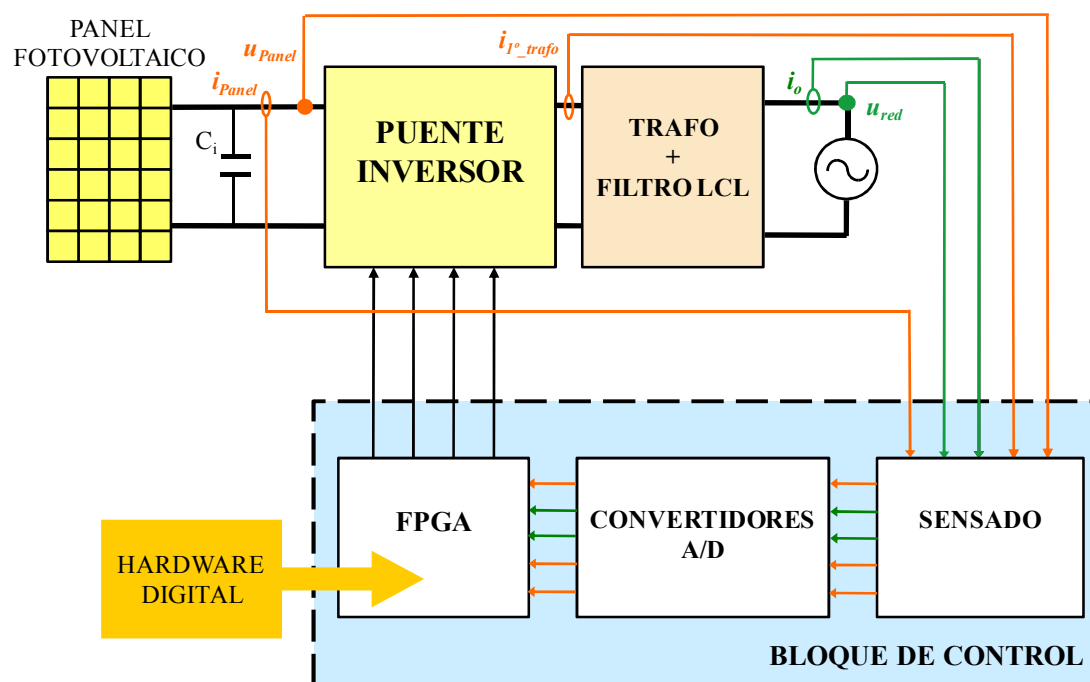


Figura 4.1. Situación del hardware digital dentro del diagrama de bloques del sistema completo.

4.1. EL CONTROL DEL PUENTE INVERSOR EN BUCLE ABIERTO.

El control en bucle abierto (sin conexión a red, *Figura 4.4*) implementado en este sistema tiene dos finalidades principalmente:

- Validar el funcionamiento de la etapa de potencia y de los circuitos auxiliares (fuentes de alimentación).
- Caracterizar la etapa de potencia: Tensiones y corrientes máximas, rendimiento, medidas EMI y ajuste de algunos de los parámetros de funcionamiento (tiempos muertos entre los disparos de los transistores de la misma rama).

En el funcionamiento en bucle abierto, el control del inversor se realiza mediante una modulación PWM unipolar cuyo esquema de funcionamiento es el que se muestra en la *Figura 4.2*:

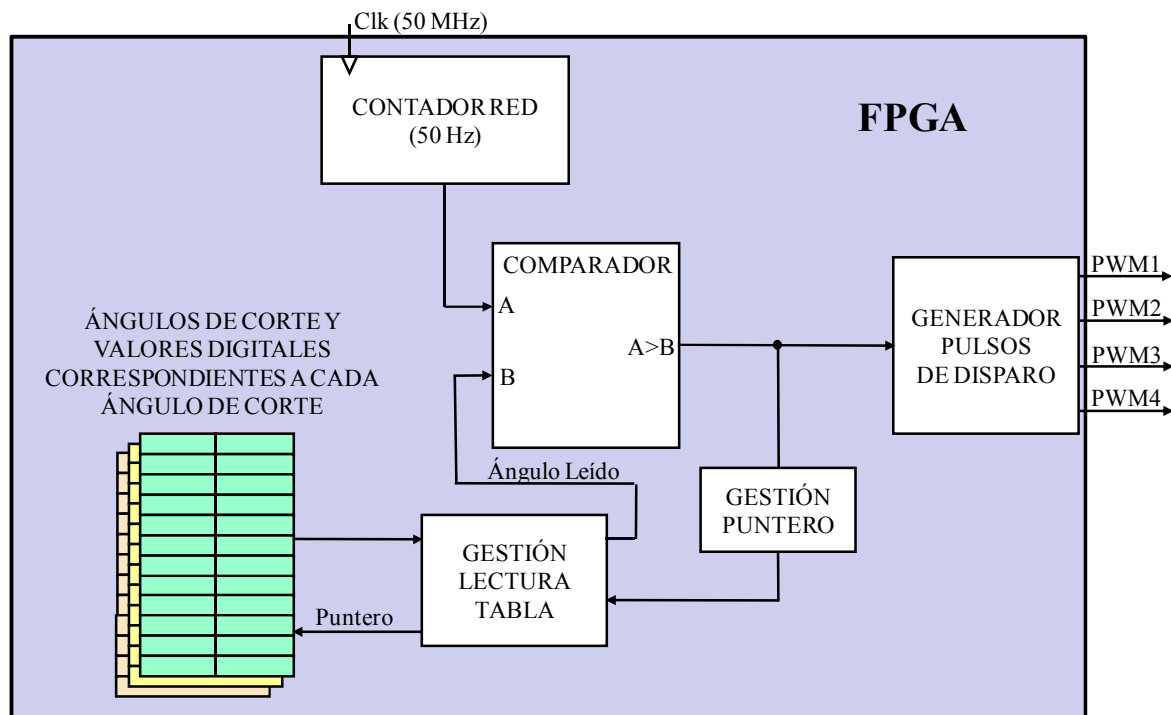


Figura 4.2. Esquemático del bloque de control del puente inversor en bucle abierto sintetizado en la FPGA

El funcionamiento de este bloque de control se basa en la comparación de un contador cuya cuenta máxima finaliza en cada ciclo de red, con otro número cuyo valor está asociado a un ángulo de disparo. En función de esta comparación se generan los pulsos de disparo de los transistores y además, se gestiona un puntero para la lectura de unas tablas. En cada tabla se almacenan los ángulos de disparo y los valores digitales que corresponden a cada ángulo de disparo (ver *Figura 4.3*), disponiendo en este caso de 3 tablas que permiten la elección de diferentes condiciones de funcionamiento (frecuencia de conmutación e índice de modulación). El hardware de control lee las 3 tablas en cada ciclo de reloj y genera los disparos en función de la tabla seleccionada en cada instante. Para ello, hay un puntero que incrementa su valor una unidad cada vez que el valor del contador de red supera el valor del ángulo leído, pasando a comparar el siguiente ángulo de disparo.

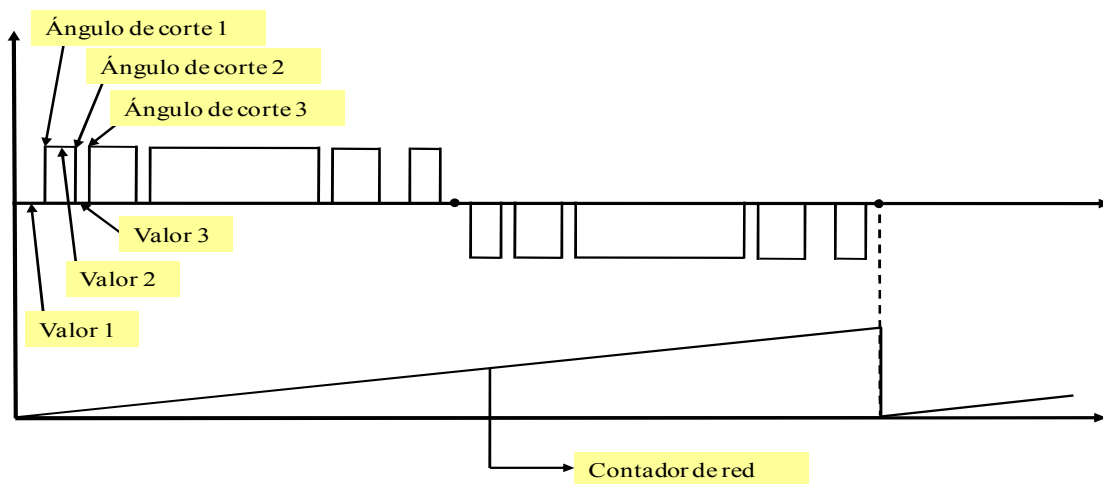


Figura 4.3. Representación de la PWM obtenida a partir de la comparación del contador de red y los ángulos de corte.

El método desarrollado para facilitar la creación de tablas de disparo y la reprogramación de la FPGA está representado en la *Figura 4.4*.

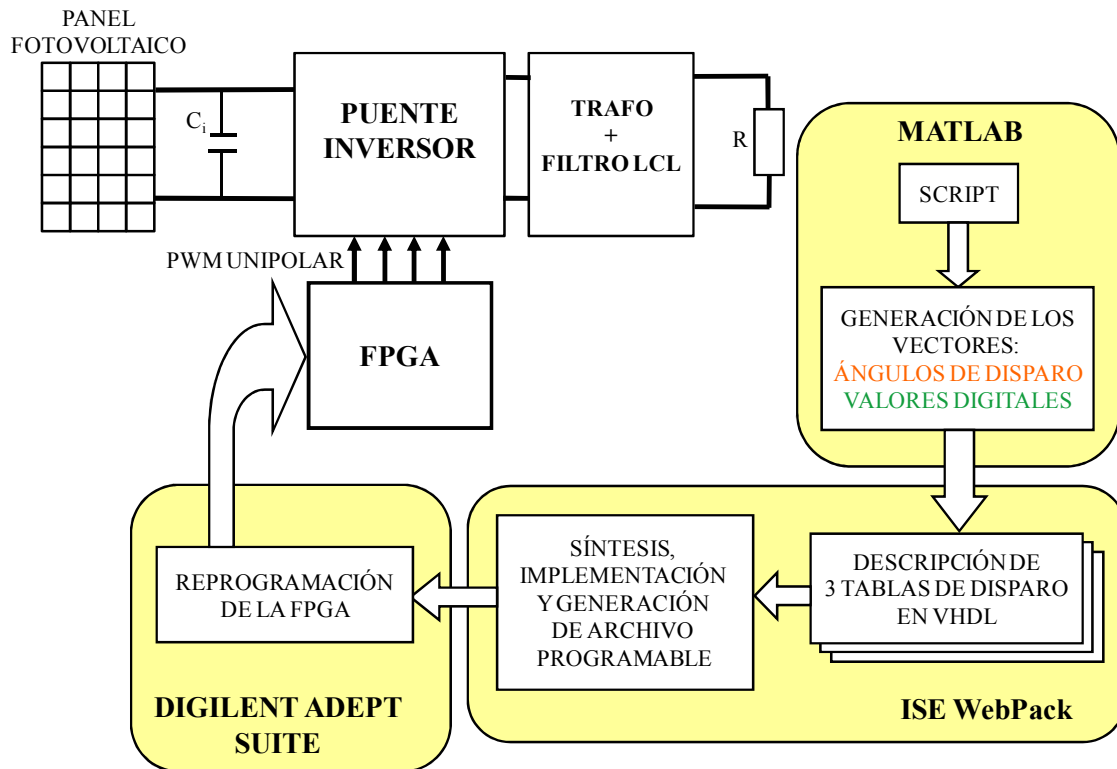


Figura 4.4. Esquema del proceso de reprogramación de la FPGA con la PWM unipolar.

Este método consiste en seguir los pasos que se describen a continuación:

1. Se ha creado un pequeño *script* que genera unas tablas con los ángulos de disparo y los estados digitales correspondientes a cada ángulo de disparo (Ver *ANEXO V*). Para ello se ha empleado el software matemático basado en lenguaje de alto nivel, MATLAB. El diseño de este *script* se centra en la generación de una señal PWM para un puente completo de transistores, conmutando una rama a alta frecuencia y la otra a baja frecuencia en cada semiciclo de red; y alternando las ramas en cada semiciclo. Las opciones de configuración que permite el script para la generación de la PWM son la frecuencia de conmutación y el índice de modulación. Por otra parte, la resolución es función de la frecuencia de reloj de la FPGA (50 MHz) y de la frecuencia de la red (50 Hz), con lo que en este caso se tiene una resolución de $1 \cdot 10^6$ puntos por periodo de red.
2. Se ha realizado, empleando código VHDL, un conjunto de tablas que contienen los paquetes de datos generados por el script. Estas tablas pueden modificarse



fácilmente, ya que disponen de un espacio específico para la introducción de los ángulos de disparo y las señales digitales de disparo correspondientes. De esta manera, se minimiza el tiempo de reprogramación y síntesis de la FPGA a la hora de realizar las pruebas con diferentes tablas.

3. Tras la generación de las 3 tablas con sus correspondientes paquetes de datos, se realiza la síntesis del diseño, implementación del mismo y generación del archivo programable. Para ello se hace uso de la herramienta de descripción de circuitos digitales ISE WebPack, de XILINX.
4. El último paso es el proceso de reprogramación de la FPGA, que en este caso se lleva a cabo con el software específico de DIGILENT denominado Digilent Adept Suite.

4.1.1. SIMULACIÓN DEL BLOQUE DE DISPARO DEL INVERSOR EN BUCLE ABIERTO.

Antes de comprobar experimentalmente el correcto funcionamiento del bloque de control del puente inversor en lazo abierto, se ha comenzado con una simulación del mismo a través de la herramienta de simulación de circuitos digitales, MODELSIM. Para ello, se ha diseñado en primer lugar un banco de pruebas con el que poder realizar una primera simulación de este bloque de disparos. A su vez, se ha incluido un modelo de inversor con filtro diseñado en VHDL, necesario para poder llevar a cabo dicha simulación. Ambos bloques se describen en el *ANEXO VI*, en el apartado de código simulable.

En la *Figura 4.5* se representa el banco de pruebas empleado para la simulación del inversor en lazo abierto, que está formada por estos 2 elementos:

- El bloque de generación de la PWM, que realiza el disparo de los transistores a partir de la lectura de una serie de tablas donde están almacenados los ángulos de disparo y los valores de disparo
- El modelo de inversor, que simula el comportamiento de un inversor monofásico. Las señales de entrada de este inversor son, además de los pulsos de los transistores, la tensión de entrada (U_e), y la tensión de red (U_{RED}). Entre las señales de salida,

este modelo de inversor permite visualizar la corriente de entrada (I_e) y salida del mismo (I_o), y la tensión de salida (U_{ab}). Además, este inversor dispone entre sus parámetros de configuración, de la inductancia (L) y la resistencia (R) de salida propias de un inversor.

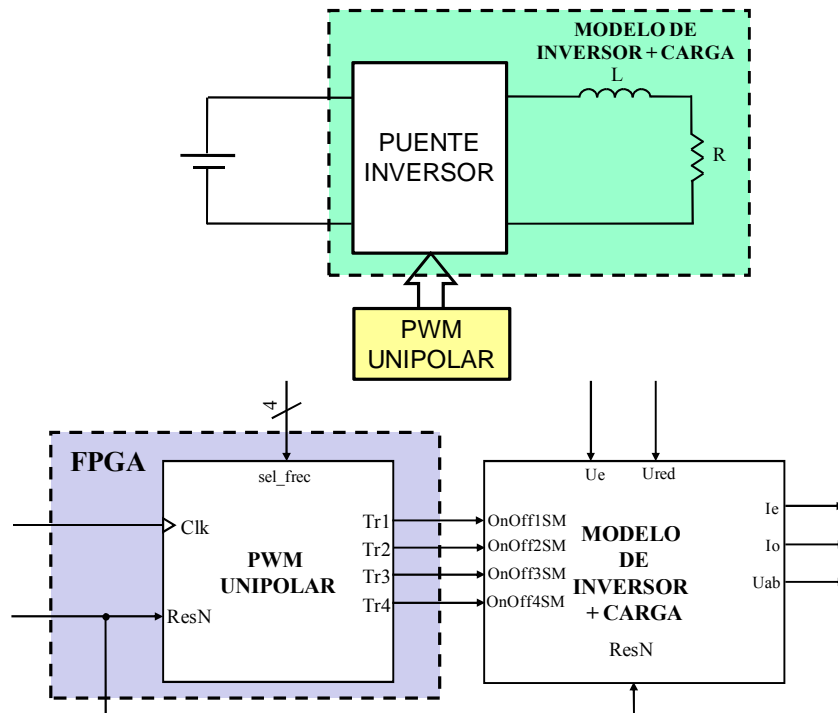


Figura 4.5. Banco de pruebas empleado para la simulación del inversor en lazo abierto.

Con el fin de representar varias PWM con diferente frecuencia de simulación, se ha creado la simulación que se muestra en la Figura 4.6, donde en función de un selector de frecuencia, el sistema carga diferentes tablas de disparo. La tensión de entrada del lado de continua que se ha escogido para esta simulación es de 100 V. Por otra parte se han empleado como parámetros de carga una $L = 800 \mu\text{H}$ y una $R = 1\Omega$. En estas condiciones de funcionamiento, las frecuencias de conmutación que se representan son de 1.5 kHz, 3 kHz y 10 kHz. A la vista de las corrientes de entrada y salida del inversor, así como la tensión de salida del inversor, se puede apreciar claramente la diferencia de las conmutaciones según se selecciona una frecuencia de conmutación u otra.

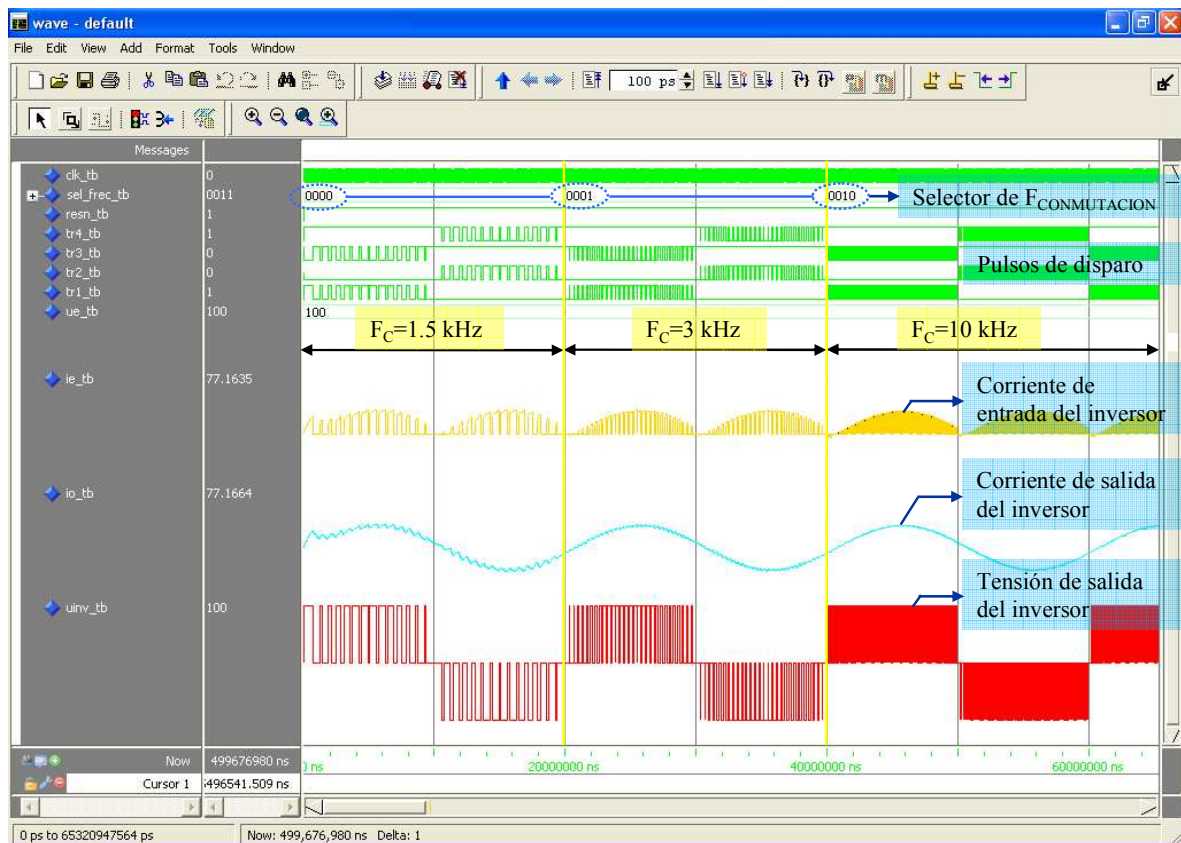


Figura 4.6. Simulación en VHDL del control del puente inversor en lazo abierto con frecuencias de conmutación de 1.5 kHz, 3 kHz y 10 kHz..

4.1.2. VALIDACIÓN EXPERIMENTAL DEL BLOQUE DE DISPARO DEL INVERSOR EN BUCLE ABIERTO.

El control diseñado para el puente inversor en lazo abierto va a permitir comprobar el correcto funcionamiento de la etapa de potencia de manera independiente con respecto al resto del sistema de control (los sensores de medida y la tarjeta de conversión analógica-digital).

Los ensayos realizados con el prototipo de pruebas en bucle abierto aportan además información que permiten conocer las mejores condiciones de funcionamiento del sistema real:

- Por un lado, se tratará de optimizar el tiempo muerto, de manera que se reduzcan los glitches durante las conmutaciones obteniendo así un rendimiento del sistema lo

más elevado posible. Este análisis solo puede realizarse experimentalmente, pues depende en gran medida del comportamiento real de los transistores, lo cual no se tiene en cuenta en la simulación.

- Por otra parte, los ensayos experimentales sirven para comprobar el aguante real de la etapa de potencia diseñada en este prototipo.

Para la realización de estas pruebas se ha empleado como fuente de entrada del puente inversor, una fuente de continua modelo 6015A de HP/AGILENT, capaz de dar entre 0 y 500 V, de 0 a 5 A y una potencia total de hasta 1 kW. Ver *Figura 4.7*.



Figura 4.7. Fuente de alimentación modelo 6015A, de HP/Agilent.

Las pruebas experimentales del inversor en lazo abierto se han realizado empleando el conjunto de transformador + filtro LCL diseñado para el proyecto UNISOL, con una carga resistiva de 165Ω a la salida. Estas pruebas se han realizado con diferentes frecuencias de conmutación, índices de modulación y tiempos muertos, habiendo logrado los mejores resultados con una $F_{\text{CONMUTACIÓN}} = 30 \text{ kHz}$, índice de modulación $m_a = 0.8$ y un tiempo muerto de $1 \mu\text{s}$. El resultado gráfico de esta prueba se puede observar en la *Figura 4.8*, donde se representan la tensión de salida del inversor y la corriente circulante por la carga resistiva. En esta figura se puede apreciar que los picos típicos de las conmutaciones son despreciables respecto al valor de tensión de trabajo, que este caso es de $131 V_{\text{eficaces}}$ en la carga.

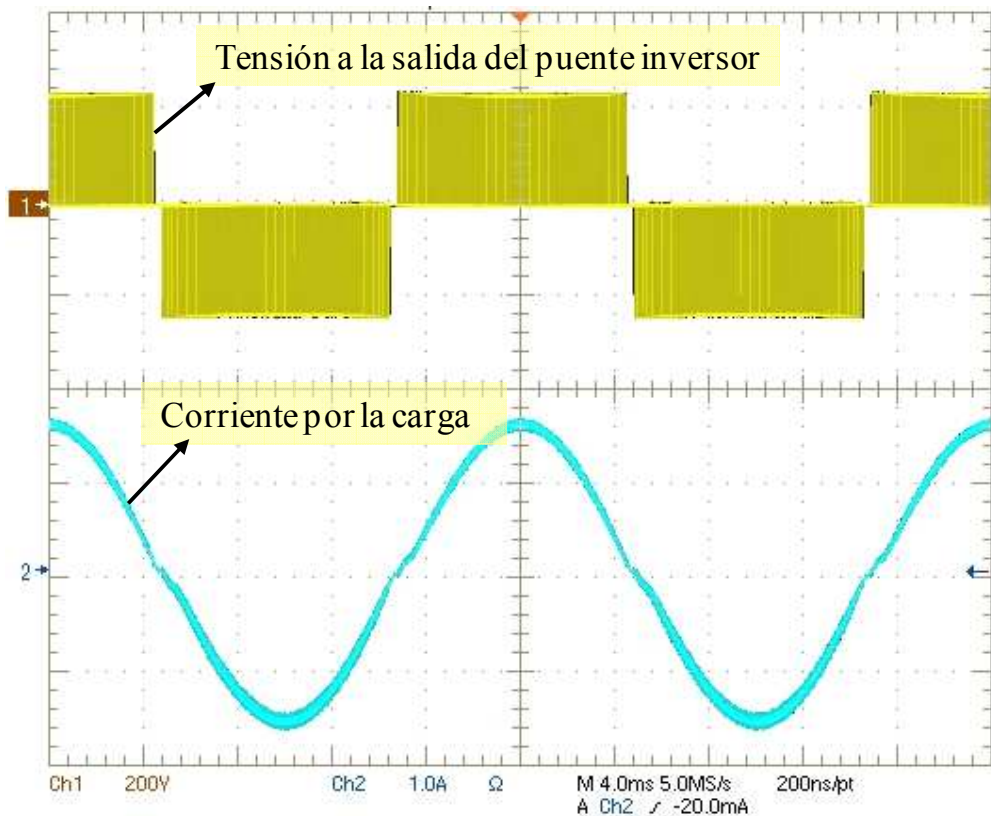


Figura 4.8. Corriente de salida del inversor monofásico en bucle abierto, empleando $F_{CONMUTACIÓN} = 30 \text{ kHz}$, $m_a = 0.8$ y $t_{MUERTO} = 1 \mu\text{s}$.

En cuanto a la capacidad de aguante de los transistores, se ha realizado una prueba consistente en emplear diversas cargas resistivas, aumentando progresivamente la tensión de entrada del sistema hasta tener una tensión en la carga de $220 \text{ V}_{\text{eficaces}}$, correspondientes a la tensión de red. La realización de estas pruebas se ha llevado a cabo empleando la PWM con una frecuencia de conmutación de 30 kHz , un índice de modulación de 0.8 y un tiempo muerto entre conmutaciones de $1 \mu\text{s}$.

En la *Tabla 4.1* se recogen los resultados más importantes de estos ensayos experimentales, donde se representa el rendimiento del inversor en bucle abierto al emplear 2 cargas resistivas diferentes. La primera de ellas es una resistencia de 165Ω , lo que permitiría una potencia de salida de hasta $P_o = V^2 / R = 220^2 / 165 = 300 \text{ W}$, mientras que la otra, con un valor de 82.5Ω , supondría un máximo de $P_o = V^2 / R = 220^2 / 82.5 = 600 \text{ W}$.



CARGA (Ω)	V_{PANEL} (V)	I_{PANEL} (A)	P_{PANEL} (W)	V_o (V)	I_o (A)	P_o (W)	Rendimiento (%)
165	50	0.32	16	47.8	0.31	14.6	91.25
	70	0.45	31.5	67	0.41	27.5	87.3
	100	0.64	64	95.5	0.58	56.6	88.4
	150	0.96	144	143.3	0.88	127.8	88.75
	170	1.08	183.6	161.9	0.99	163	88.77
	200	1.27	254	190.2	1.17	225.4	88.74
	232	1.47	341	220	1.35	301.3	88.35
82.5	100	1.18	118	89	1.16	103.3	87.5
	150	1.74	261	132	1.68	226.7	86.8
	175	2.02	353.5	154	1.94	303.2	85.8
	200	2.29	458	174	2.17	382.3	83.5

Tabla 4.1. Rendimiento del inversor en lazo abierto, empleando $F_{\text{CONMUTACIÓN}} = 30 \text{ kHz}$ y $t_{\text{muerto}} = 1 \mu\text{s}$.

De los resultados mostrados en la Tabla 4.1 hay que destacar:

1. En primer lugar, que no se ha podido llegar a entregar más de 400 W a la carga, ya que el inversor no lo soporta, rompiéndose los principales componentes del mismo, que son los drivers y los transistores. Entre las posibles causas se puede destacar el efecto Miller que aparece debido a la capacidad interna existente entre el drenador y la puerta de los transistores. Esto hace que con altas derivadas de tensión se dispare el transistor cuando no debe, produciendo un cortocircuito en la rama correspondiente y su destrucción. En cualquier caso, este defecto de funcionamiento queda por estudiar, ya que no forma parte del objetivo principal de este proyecto.



2. En segundo lugar, se puede destacar que el rendimiento obtenido en lazo abierto para el inversor, entregando potencias máximas de 400 W, llega a un valor aceptable de 83.5 %.

CONCLUSIONES DE LAS PRUEBAS DEL INVERSOR EN LAZO ABIERTO

De la realización de las pruebas del inversor monofásico en lazo abierto se han extraído las siguientes conclusiones:

- Se ha validado experimentalmente el funcionamiento de la etapa de potencia diseñada para el prototipo de pruebas del proyecto UNISOL.
- En los ensayos experimentales, la máxima potencia extraída del convertidor ha sido 400 W con carga resistiva y a partir de esta potencia los transistores del puente inversor experimentaban un mal funcionamiento y se destruían.
- Considerando que el puente inversor solo es capaz de suministrar 400 W y suponiendo un rendimiento aproximado de 85%, los 60 W de pérdidas estarían concentrados principalmente en los transistores. Este es un valor soportable por el disipador empleado en este sistema, que se ha sobredimensionado para poder evacuar más de 200 W.

4.2. EL CONTROL DE LOS CONVERTIDORES A/D.

La tarjeta de los ADC's tiene el objetivo de muestrear los distintos parámetros eléctricos medidos por los sensores, convirtiéndolos a datos binarios que son enviados en serie hacia la FPGA. Ésta, a su vez, es la encargada de registrar y gestionar correctamente los datos recibidos para generar los disparos del inversor.

El módulo de control de los ADC's se ha implementado en la FPGA junto con el bloque destinado al control en lazo cerrado, ya que es indispensable la digitalización de las magnitudes eléctricas para que la FPGA pueda gestionar los datos y generar los disparos correctamente. Ver *Figura 4.9*.

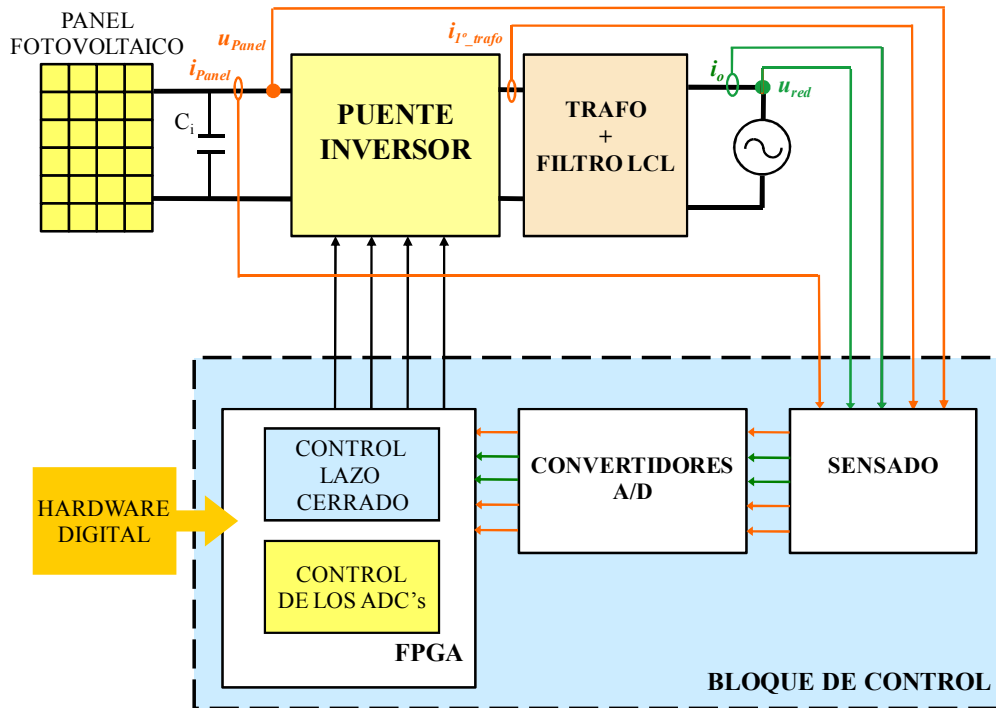


Figura 4.9. Diagrama de bloques del prototipo de pruebas con esquema básico del hardware digital.

El convertidor A/D serie empleado en este proyecto es el modelo ADS 7887 descrito anteriormente, y cuyo cronograma de tiempos de funcionamiento se muestra en la Figura 4.10.

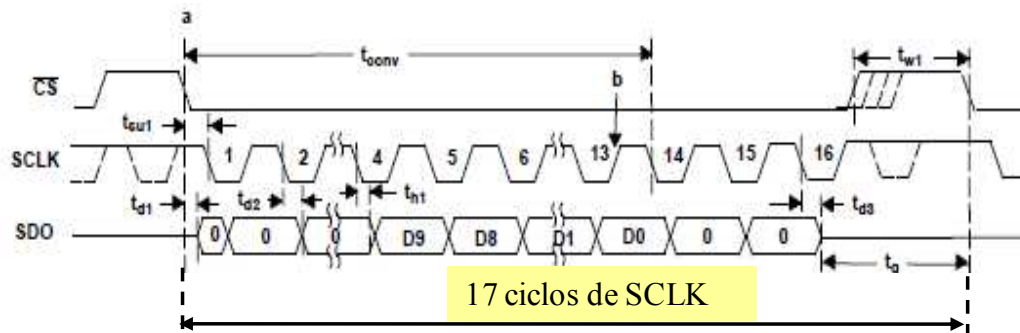


Figura 4.10. Cronograma de tiempos para el funcionamiento del convertidor ADS7887.

El principio de funcionamiento del convertidor A/D consiste en que cada vez que se produce un flanco de bajada en la señal de control denominada Chip Select (\overline{CS}), se produce el muestreo de un dato analógico y comienza el proceso de conversión. Cada dato es convertido en una cadena de bits que es enviada en serie por la salida digital del



ADC (SDO) hacia la FPGA. El proceso de envío de bits está sincronizado con una señal de reloj externa (SCLK), que en este caso es generada a partir de la FPGA, al igual que la señal Chip Select. Cada cadena de bits enviada por el ADC está constituida por 4 ceros de cabecera, seguidos 10 bits que corresponden a la conversión de un dato analógico, y otros 2 ceros de cola. El proceso de conversión completo dura 17 ciclos de reloj, de los cuales 16 ciclos se emplean para la captura de la cadena de bits completa, y el ciclo restante corresponde a la activación del flanco de bajada de la señal Chip Select.

La velocidad de conversión del ADC depende de la señal de reloj empleada para la captura de cada dato. En este proyecto, se ha destinado para este fin el propio reloj interno de la FPGA, de 50 MHz y se ha dividido en frecuencia, obteniendo una señal de reloj de 12.5 MHz para la conversión de datos del ADC. De este modo, y teniendo en cuenta que es necesario un ciclo de reloj extra por cada dato convertido para volver a activar la señal CS (en total 17 ciclos), la frecuencia de muestreo del ADC es de 735.294 kHz.

El hardware digital implementado en la FPGA para la gestión de los ADC tiene dos funciones:

- Por una parte, controla el proceso de conversión de cada ADC, permitiendo que el muestreo de las diferentes señales analógicas muestreadas esté sincronizado.
- En segundo lugar, se encarga de capturar y almacenar cada cadena de 10 bits correspondiente al dato analógico muestreado, quedando a disposición de la FPGA para las posteriores operaciones.

En la *Figura 4.11* se representa el hardware digital implementado en la FPGA para el control de los ADC's, donde se pueden observar las principales señales de control descritas anteriormente y su conexión con el resto de elementos del sistema. Este bloque de control es capaz de gestionar las 5 señales digitales procedentes de la tarjeta de conversión A/D (SDO1,..., SDO5). Además, el circuito dispone de la entrada Conv_ST que sirve para que, a través de un switch de la propia FPGA, el usuario pueda iniciar el proceso de conversión manualmente. Entre las salidas existentes, se encuentran los registros de 10 bits correspondientes a cada canal de conversión de los

ADC's. En este aspecto, cabe destacar que para controlar el inversor en lazo cerrado de este proyecto, tan solo son necesarias las medidas de 2 parámetros eléctricos, por lo que solo se emplean las salidas correspondientes a 2 registros de datos de 10 bits de los 5 incluidos en este bloque. Por otra parte, la salida Conv_Ok genera una señal que hace que se ilumine un led de la tarjeta de evaluación de la FPGA cada vez que se realiza una conversión de un dato. Como la frecuencia a la que esto ocurre es superior a los 700 kHz, el parpadeo es imperceptible por el ojo humano, creando la sensación de que dicho led está iluminado constantemente.

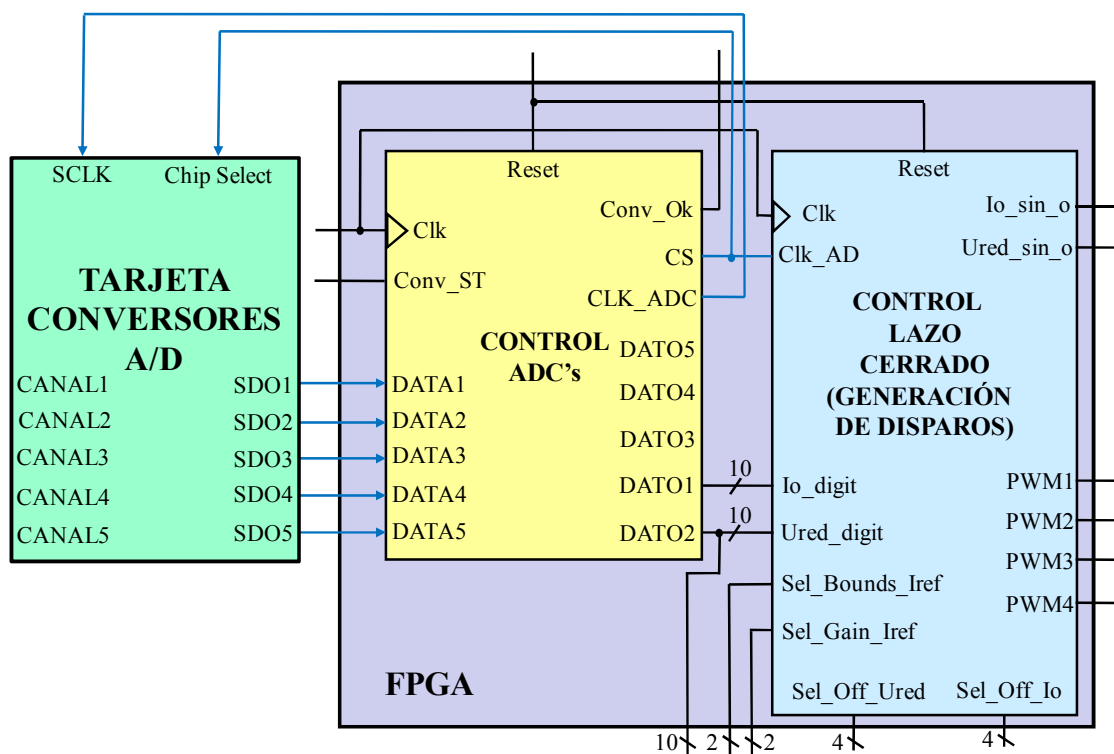


Figura 4.11. Implementación del bloque de control de los ADC's y su conexión básica.

La descripción del proceso que gestiona la tarjeta de los ADC se ha llevado a cabo con la herramienta de diseño y simulación de circuitos digitales, ISE WebPack, de XILINX. Para controlar cada convertidor A/D se ha diseñado, mediante lenguaje VHDL, una máquina de estados (ver Figura 4.12). El funcionamiento de la máquina de estados es muy sencillo, ya que dispone de tan solo 2 estados y una señal de control para pasar de uno a otro, que se denomina Conv_ST. Esta señal está asociada a un interruptor propio de la tarjeta de evaluación de la FPGA, que al activarse (ponerse a

nivel alto) realiza de forma síncrona el paso de un estado a otro. En concreto, se pasa del estado de reposo, denominado “ESPERA” al estado en el que se realiza la captura de los datos transmitidos en serie por el convertidor A/D, denominado “CAPTURA”. Durante el estado CAPTURA, la señal de control Chip Select se activa periódicamente y los bits son enviados en serie desde el ADC hacia la FPGA, almacenándose en el registro de 10 bits correspondiente. El envío en serie de los bits está sincronizado con la señal SCLK, al igual que la señal Chip Select.

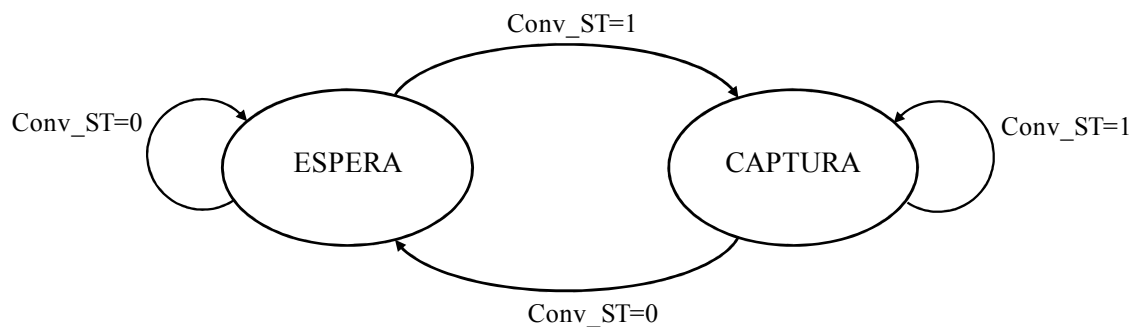


Figura 4.12. Máquina de estados para el control de los convertidores A/D.

Al finalizar el envío de cada dato, la FPGA manda desactivar el Chip Select del ADC durante un ciclo de SCLK para volver a activarse en el siguiente y comenzar así una nueva conversión. De aquí que la frecuencia de muestreo del ADC sea de 735.294 kHz, ya que se necesitan 17 ciclos de la señal de reloj del ADC para hacer una conversión ($F_{CLK_ADC} = 12.5$ MHz).

4.2.1. SIMULACIÓN DEL BLOQUE DE CONTROL PARA EL CONVERTIDOR A/D.

Para comprobar el correcto funcionamiento del bloque de control de los convertidores A/D, se ha creado un banco de pruebas en VHDL con el software de ISE WebPack (Ver Figura 4.13). El bloque de control de los ADC's comienza el proceso de conversión al activarse la señal Conv_ST. En este instante, se pasa del estado de inactividad, denominado ESPERA al estado de conversión o CAPTURA. Por otra parte, se ha generado en el banco de pruebas una cadena de bits a modo de salida digital de uno de los convertidores A/D. Mientras que el estado CAPTURA está activo, se

capturan y registran los bits, permitiendo la visualización del mismo a través de la salida DATO1. Así mismo, se puede visualizar la generación de las señales de salida CS y SCLK.

En el *ANEXO VII* se incluye la descripción VHDL de este banco de pruebas.

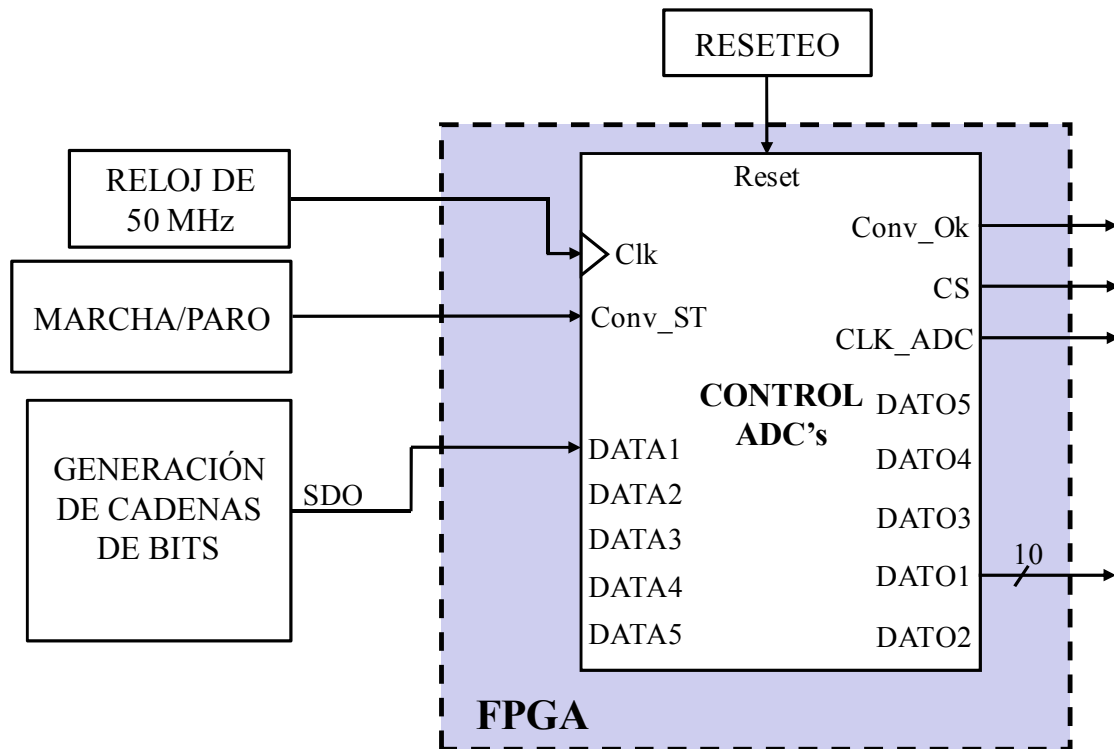


Figura 4.13. Banco de pruebas para la simulación del hardware digital del control de los ADC's.

Una vez generado el banco de pruebas completo, el diseño se simula con la herramienta de simulación denominada MODELSIM, que es lanzada a través del propio ISE WebPack, de XILINX.

En la *Figura 4.14* se muestra el resultado de la simulación, donde se pueden observar las principales señales que forman parte del control del convertidor A/D. En esta simulación se puede observar como, al activar la señal Conv_ST, se pasa del estado ESPERA al estado CAPTURA. Al realizarse dicha transición, se da la orden de activar por flanco de bajada y de forma síncrona, la señal Chip Select del ADC (CS), que inicia el proceso de conversión. Por otra parte, el proceso de generación de cadenas de bits

comienza a enviar en serie los paquetes de datos a través de la señal sdo. A medida que la cadena de bits es enviada en serie hacia la FPGA, va quedando almacenado en “registro” (sin contar los ceros de cabecera y de cola). La operación de captura del dato correspondiente a cada paso de muestreo, es almacenado en la FPGA en el registro denominado “dato” (corresponde a la salida DATO1), quedando guardado hasta el siguiente paso de muestreo. Esta operación se repite cada 17 ciclos, ya que el Chip Select es puesto a nivel alto justo al terminar de enviar el último bit el ADC (16 ciclos de reloj del ADC), volviendo a activarse por flanco de bajada en el siguiente ciclo de reloj del ADC.

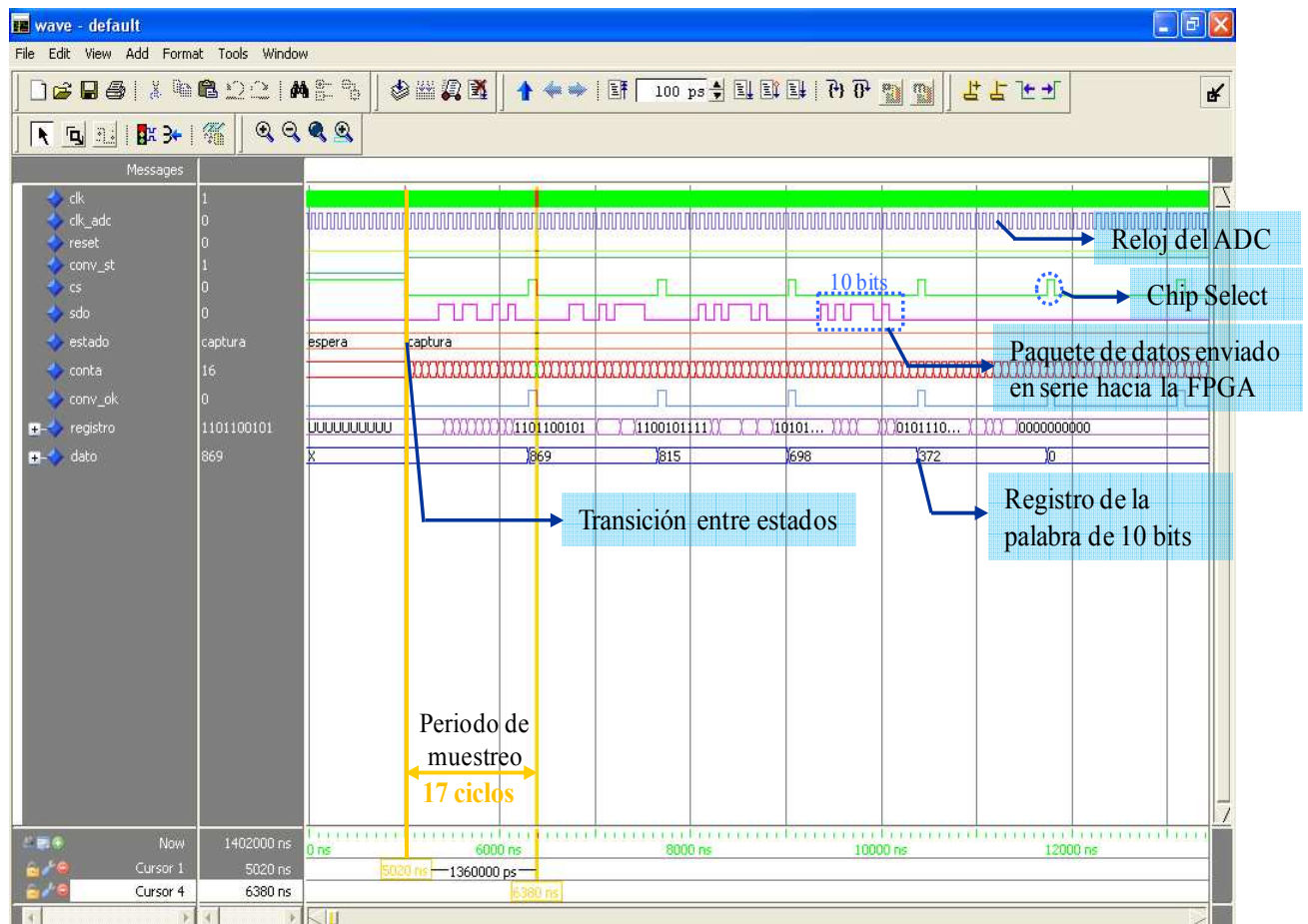


Figura 4.14. Simulación en VHDL del bloque de control para el convertidor A/D.

4.2.2. VALIDACIÓN EXPERIMENTAL DEL BLOQUE DE CONTROL PARA EL CONVERTIDOR A/D.

La validación experimental del hardware digital diseñado para el control de los ADC's tiene 2 objetivos fundamentales:

- Por una parte, permite comprobar que los resultados obtenidos mediante simulación son correctos.
- Por otra parte, sirve para validar el diseño de la tarjeta de conversión A/D.

Tras la simulación del bloque de control referente al convertidor A/D, es necesaria una validación experimental que confirme los resultados obtenidos en simulación. En este aspecto, se ha realizado el montaje que se representa en la *Figura 4.15*, que está constituido por un generador de señales, la tarjeta de conversión de los ADC's y la FPGA que contiene el circuito digital destinado al control de los ADC's.

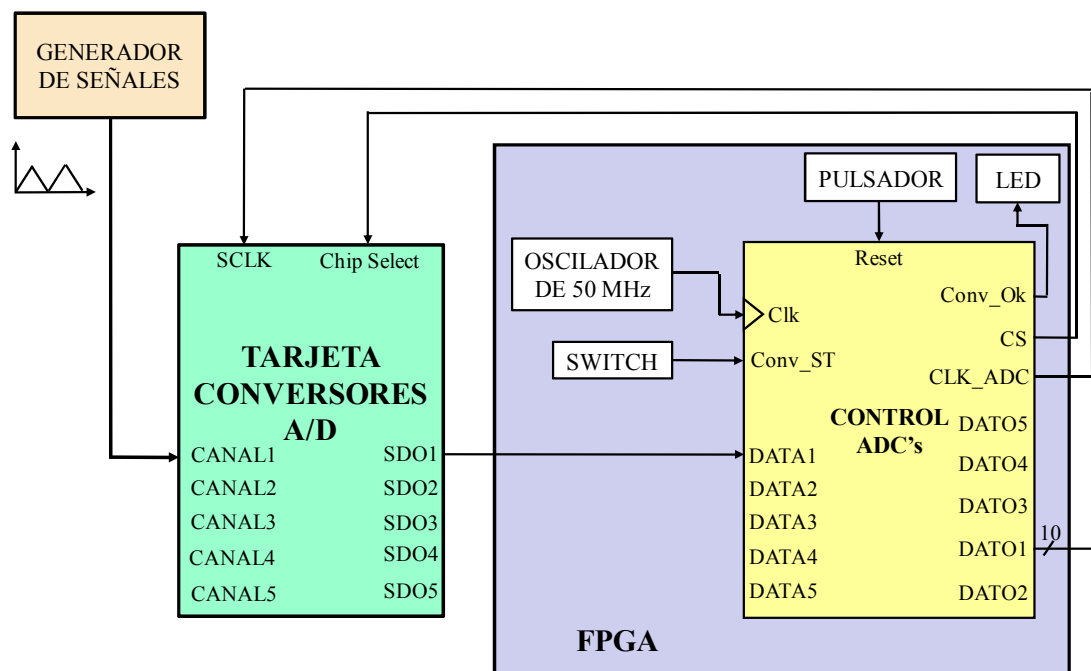


Figura 4.15. Montaje realizado para la validación experimental del hardware digital de control de la tarjeta de los ADC's.



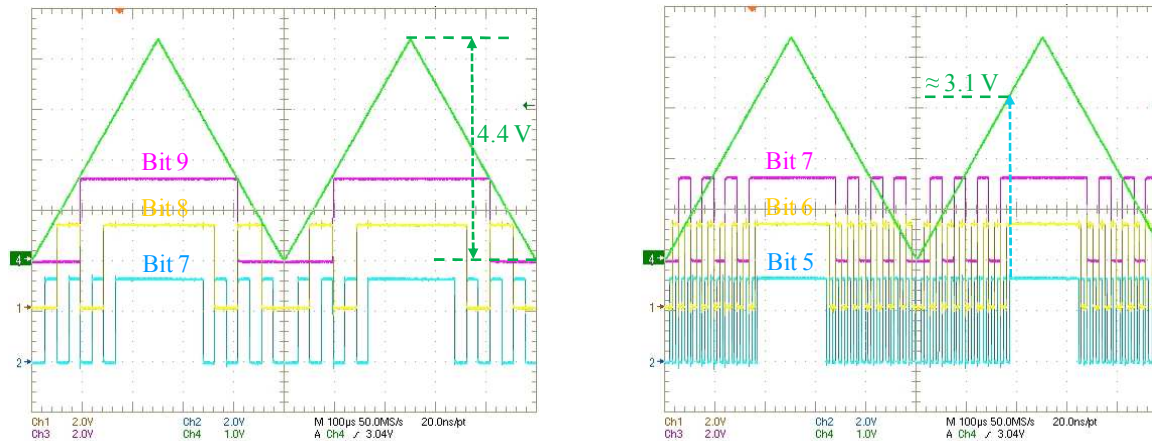
Se ha empleado para comprobar los resultados una señal triangular de 2 kHz y 4.4 V de amplitud, obtenida mediante el generador de señales. A esta señal se le ha añadido un offset de 2.2 V, necesario para que las medidas tomadas sean positivas, ya que el ADC es unipolar, por lo que no puede convertir valores inferiores a 0 V.

Para comprobar que el conjunto formado por el bloque de control de los ADC's y la tarjeta conversión A/D funcionan correctamente, se han de tener en cuenta las siguientes consideraciones:

- 1) La resolución que ofrece cada bit del ADC es $\text{Res}_{\text{BIT}}=3.223 \text{ mV/LSB}$, que se obtiene de dividir la tensión de referencia, en este caso $V_{\text{REF}}=3.3 \text{ V}$, entre la resolución propia del convertidor A/D, que es de $2^{10}=1024$.
- 2) La reconstrucción de la señal analógica a partir de los valores digitales de cada bit se obtiene empleando en cada instante de tiempo la siguiente ecuación:

$$\text{Tensión analógica} = \text{Res}_{\text{BIT}} \times \sum_{j=0}^9 [(\text{Bit } j) * 2^j] \text{ V}$$

En la *Figura 4.16* se muestran dos capturas tomadas por el osciloscopio en las que se visualizan 2 periodos de la señal triangular obtenida con el generador de señales junto a los 5 bits más significativos de cada conversión (bits 9,8 y 7 en la *Figura 4.16.a* y bits 7, 6 y 5 en la *Figura 4.16.b*). A pesar de que sería necesario representar todos los bits para poder reconstruir la señal analógica, se puede obtener una triangular aproximada con estos 5 bits, ya que son los de mayor peso. Otra consideración a tener en cuenta es que en la parte central de la señal triangular los bits se encuentran a nivel alto, lo que significa que el ADC está enviando en ese tramo una cadena de 1's. Este hecho se debe a que la señal triangular supera en ese tramo el valor de saturación del ADC, es decir, los 3.3 V de tensión de referencia (Ver *Figura 4.16.a*).



a. Bits más significativos de la señal triangular digitalizada (bits 9, 8 y 7).

b. Bits 7, 6 y 5 de la señal triangular digitalizada.

Figura 4.16. Validación experimental del control del convertidor A/D mediante una señal triangular de 2 kHz y 4.4 Vpico.

4.3. EL CONTROL DEL INVERSOR EN LAZO CERRADO CONECTADO A RED.

Tras haber validado experimentalmente el funcionamiento del inversor en bucle abierto y el control de la tarjeta de conversión de los ADC's, el siguiente paso es el diseño integrado del sistema en lazo cerrado. Para ello, se ha añadido al hardware digital de la FPGA un módulo de control que trata las señales sensadas y digitalizadas previamente, para generar los disparos de los transistores. Ver *Figura 4.9*.

En este caso, los objetivos que se han de alcanzar con el control del sistema completo, son:

1. Asegurar que la corriente de red sea directamente proporcional a la tensión de red.
2. Asegurar que el panel fotovoltaico trabaje siempre en el punto de máxima potencia.

Para lograr estas especificaciones, se propone el sistema de control que se representa en la *Figura 4.17*, que incluye:

- Un bloque de control para asegurar el funcionamiento del panel solar en el punto de máxima potencia (MPPT).
- Un regulador de corriente por histéresis, que es el encargado de generar los disparos de los transistores para que la corriente entregada por el puente inversor siga a la corriente de referencia. La amplitud de la corriente de referencia es generada por el lazo de control externo (el lazo MPPT). El propio lazo de corriente se asegura de que la corriente esté en fase con la tensión de red y tenga su misma forma de onda.
- Una inductancia de acoplamiento con la red, que hace posible que el inversor pueda comportarse como una fuente de corriente. Adicionalmente, se puede dividir la inductancia de acoplamiento en 2 inductancias en serie con un condensador adicional, formando un filtro LCL (ver *Figura 4.17*), con el fin de reducir los armónicos de corriente inyectados a la red.

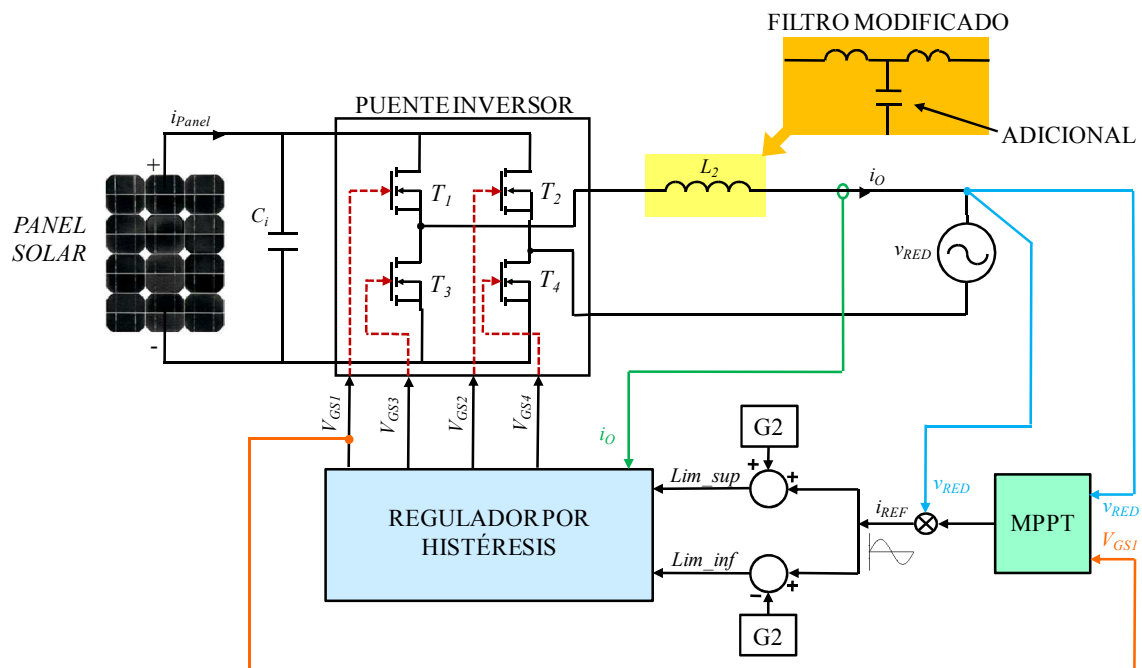


Figura 4.17. Diagrama de bloques del inversor completo, con lazo de corriente y seguimiento del punto de máxima potencia.

La metodología seguida para el diseño del control digital de este inversor monofásico es la siguiente:



- 1) Se comienza por el diseño de un lazo de control para la regulación de la corriente de salida del inversor, o lo que es lo mismo, un regulador de la potencia entregada a la red. Este lazo está basado en un control por histéresis que abre y cierra los transistores adecuadamente en función de unos límites establecidos previamente a partir de la corriente de referencia. **El objetivo principal del diseño del control en lazo cerrado del inversor es este lazo de corriente y su validación experimental.**
- 2) Se añade en segundo lugar un lazo para controlar la tensión y corriente a la entrada del inversor, de manera que los paneles fotovoltaicos trabajen en el punto de máxima potencia. El objetivo de este segundo lazo se centra en que, partiendo de las medidas de tensión de red y el ancho de los pulsos de disparo de uno de los transistores, se logre llevar el inversor a trabajar en el punto en el que se obtenga la mayor potencia del panel solar.

4.3.1. EL LAZO DE CORRIENTE DE SALIDA.

Uno de los objetivos principales de conectar un inversor a la red es el de conseguir introducir una corriente que esté en fase con la tensión de red y tenga su misma forma de onda sinusoidal. En este aspecto, el bloque de control de corriente que se propone permite generar una corriente de referencia i_{REF} a partir de la propia tensión de red, asegurando que la señal de referencia esté sincronizada con la red y tenga su misma forma de onda.

El lazo de corriente implementado para asegurar una corriente de red que esté en fase con la tensión de red se representa en la *Figura 4.18*, cuyas variables de control son la tensión de red, la corriente de salida y el pulso de uno de los transistores del inversor.

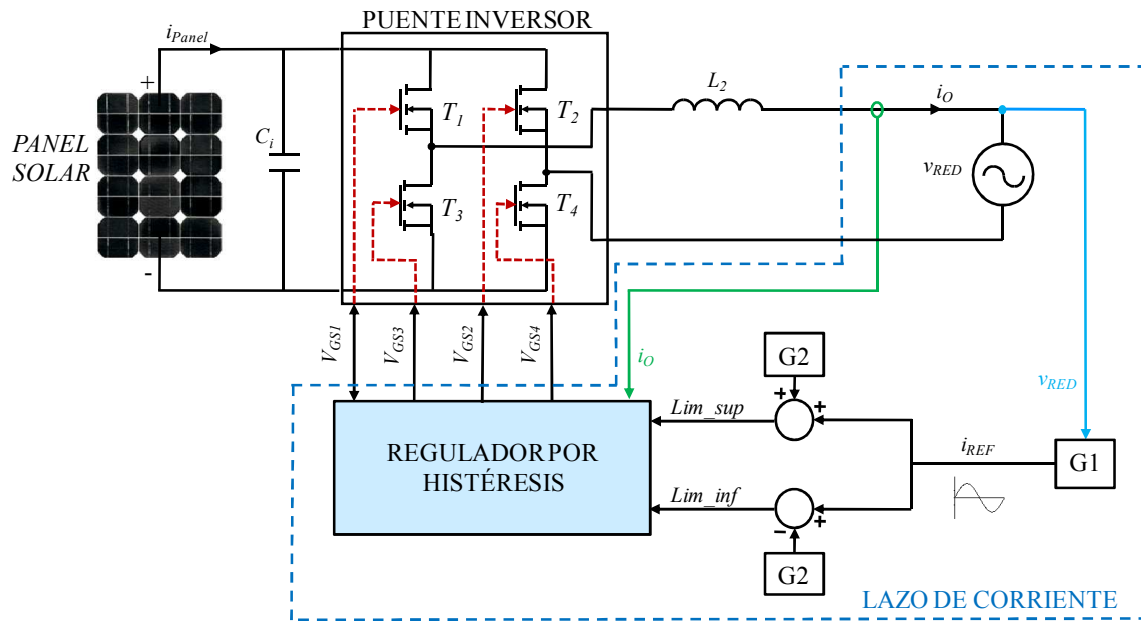


Figura 4.18. Diagrama de bloques del inversor con el lazo de corriente.

El lazo de corriente implementado en este inversor tiene la particularidad de que se controla a través de un regulador por histéresis o de tipo bang-bang. El funcionamiento de este tipo de reguladores consiste en que cada vez que la corriente de salida del inversor (i_o) rebasa unos límites establecidos, se modifica el disparo de los transistores, de manera que la corriente vuelva a estar dentro de los límites impuestos. El límite superior se calcula sumando a la corriente de referencia (i_{REF}) un valor constante, y el límite inferior se calcula restando a la i_{REF} ese mismo valor constante. Según la Figura 4.18, si T_1 y T_4 están cerrados, la corriente i_o subirá. Cuando llegue al límite superior, se activarán T_2 y T_3 , con lo que la i_o bajará hasta el límite inferior; y el ciclo comenzará de nuevo. Los límites se generan a partir de la tensión sinusoidal medida de la propia red, la cual se multiplica por una constante y se le suma y resta otra constante, obteniendo los límites superior (Lim_sup) e inferior (Lim_inf).

En la Figura 4.19 se puede ver una representación del funcionamiento del regulador de corriente por histéresis. Esta corriente está constituida por pequeñas rampas cuya pendiente es función de la tensión de carga y descarga aplicada sobre la inductancia L_2 en cada paso de muestreo. Dado que la referencia es generada a partir de la tensión de red, la forma de la corriente sigue a la referencia sinusoidal.

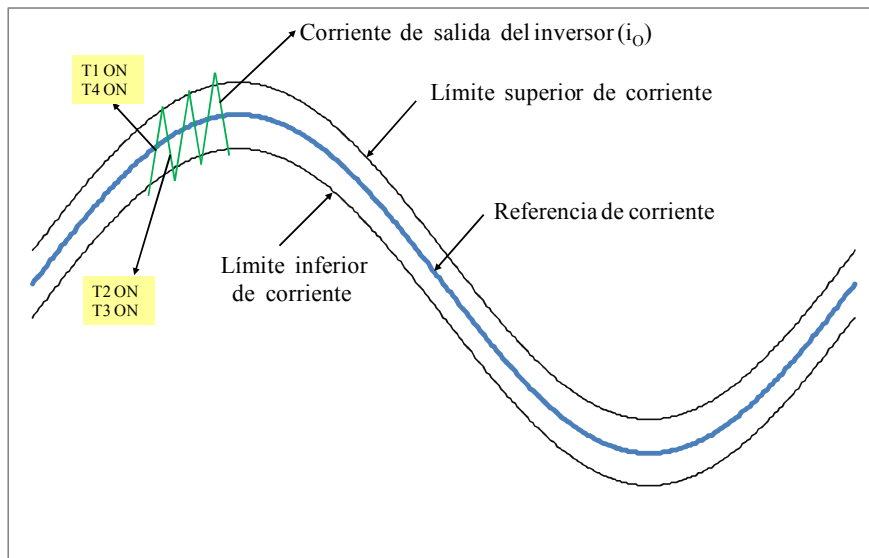


Figura 4.19. Representación del funcionamiento del regulador de corriente por histéresis.

El diagrama de estados implementado en la FPGA para el control por histéresis es el que se observa en la *Figura 4.20* y consiste en el paso de un estado UP a otro estado DOWN cuando la corriente de salida del inversor supera el límite Lim_sup . De igual modo, se pasa del estado DOWN al UP cuando la corriente de salida es inferior al límite Lim_inf .

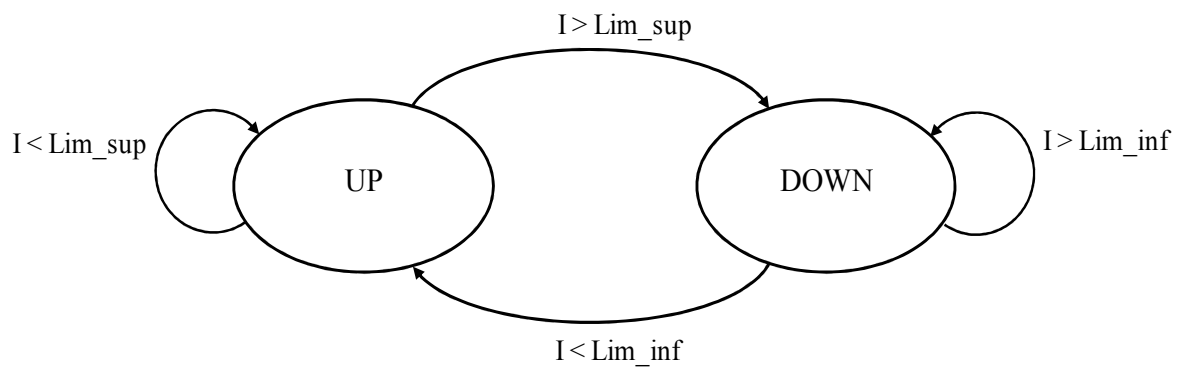


Figura 4.20. Diagrama de estados del regulador por histéresis.

El hardware digital implementado para el control del regulador de corriente por histéresis se muestra en la *Figura 4.21*, cuyo funcionamiento se describe a continuación:

1. El primer paso es el tratamiento de las señales correspondientes a la tensión de red y la corriente de salida para eliminar el offset de cada una de ellas. Este offset tiene su origen en el efecto que se incluyó analógicamente para que los convertidores A/D pudieran procesar correctamente todo el rango de valores deseado, ya que sólo son capaces de convertir tensiones de entre 0 y 2.5 V.
2. Una vez eliminado el efecto de offset, las señales quedan centradas ambas en el origen, pudiendo llevar a cabo de forma sencilla el control por histéresis. En este regulador, la corriente de referencia se obtiene directamente a partir de la tensión de red multiplicada por una constante que dependerá de la corriente que se quiere entregar a la red.
3. En cuanto al control por histéresis propiamente dicho, el elemento principal del sistema es el multiplexor, que permite, en función del estado de uno de los transistores (el transistor 1 o el 4 en este caso), seleccionar el límite superior e inferior de corriente, obligando a la corriente a que permanezca dentro del rango de valores impuesto.

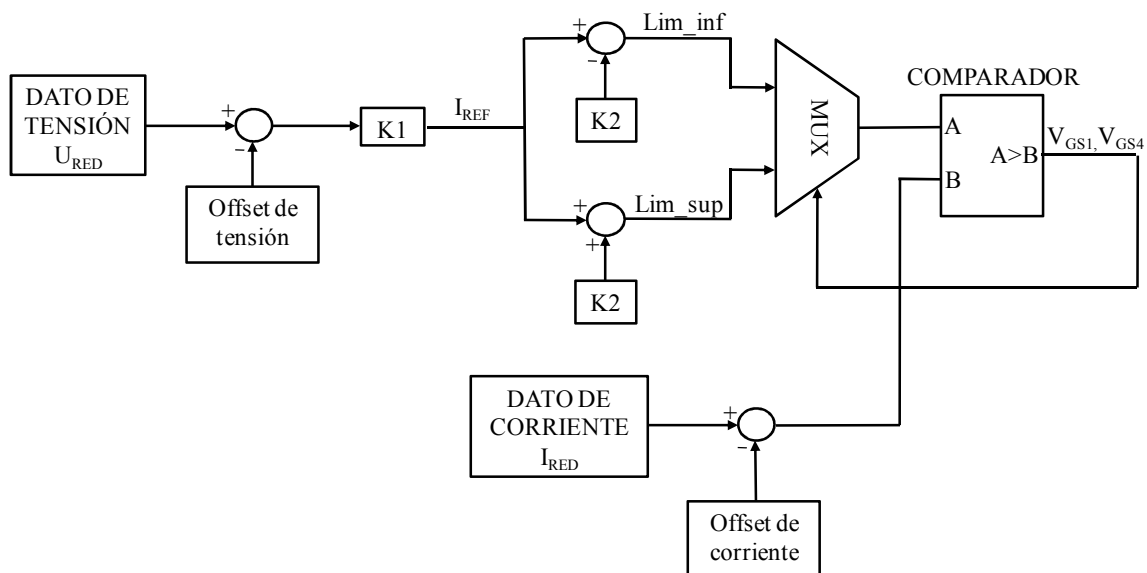


Figura 4.21. Diagrama de bloques implementado en la FPGA para el regulador por histéresis.

Una característica importante del control por histéresis es que, al ser digital, no se puede evitar que en cada paso de muestreo se sobrepasen los límites impuestos, ya que la pendiente de la corriente de salida depende de la inductancia en serie con la red



(L₂). Lo que hace el sistema de control es comprobar en cada paso de muestreo si se ha sobrepasado uno de los límites impuestos, y en caso afirmativo, corregir el comportamiento de la corriente para que vuelva a estar dentro de los límites. Es decir, la construcción de la señal de corriente a partir del regulador por histéresis no dispone de unas condiciones de frecuencia de conmutación constante, sino que se trata de una frecuencia variable que depende de si se han rebasado los límites impuestos en cada paso de muestreo. La frecuencia de muestreo en este sistema se encuentra sincronizada con la frecuencia de adquisición de datos del convertidor A/D ($F_{\text{MUESTREO_ADC}} = 735.294$ kHz).

A continuación se describe la metodología seguida para llevar a cabo el diseño y validación del funcionamiento del inversor monofásico con el lazo de corriente:

- Se ha diseñado el control digital VHDL y se ha simulado con MODELSIM, para lo cual se ha generado un banco de pruebas y un modelo digital del convertidor analógico-digital.
- Finalmente, se ha implementado el diseño del control de corriente en la FPGA y se ha validado experimentalmente de 3 formas: En cortocircuito, con una carga resistiva a la salida del inversor y directamente conectado a la red.

4.3.2. SIMULACIÓN DEL INVERSOR CON LAZO DE CORRIENTE CONTROLADO POR HISTÉRESIS.

Al igual que con el resto de circuitos digitales implementados en la FPGA, es imprescindible disponer de una simulación satisfactoria del hardware de control para el lazo cerrado. En este caso, la simulación del puente inversor en lazo cerrado se ha llevado a cabo mediante el banco de ensayos que se representa en la *Figura 4.22*.

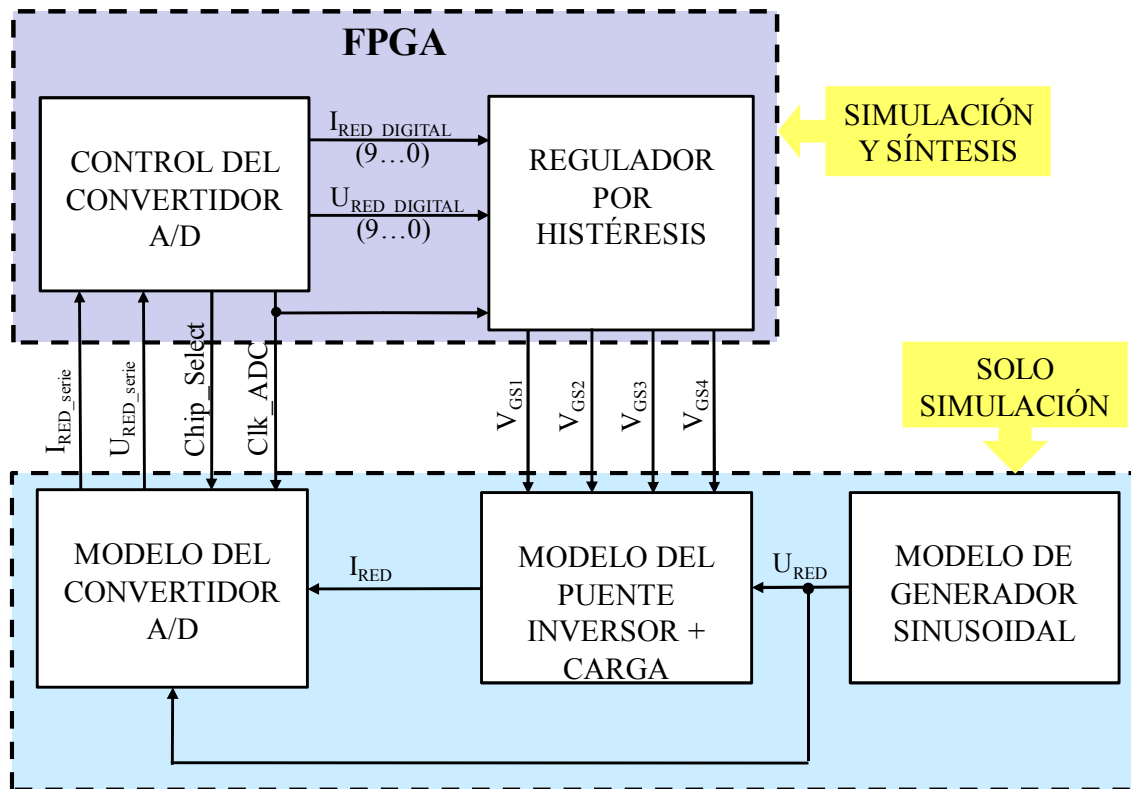


Figura 4.22. Banco de ensayos creado para la simulación del inversor con regulador por histéresis.

Los bloques que se implementan físicamente en la FPGA en este caso son:

- El control del convertidor A/D.
- El regulador por histéresis.

A estos bloques se le han añadido los siguientes solo para simulación:

- Modelo de una señal sinusoidal, que hace las veces de la tensión de red.
- Modelo de un convertidor A/D. Emula el funcionamiento de los convertidores A/D empleados en la medición de la tensión de red y la corriente de salida del inversor.
- Modelo del puente inversor. Se trata del mismo modelo que se empleó para la simulación del inversor en bucle abierto.

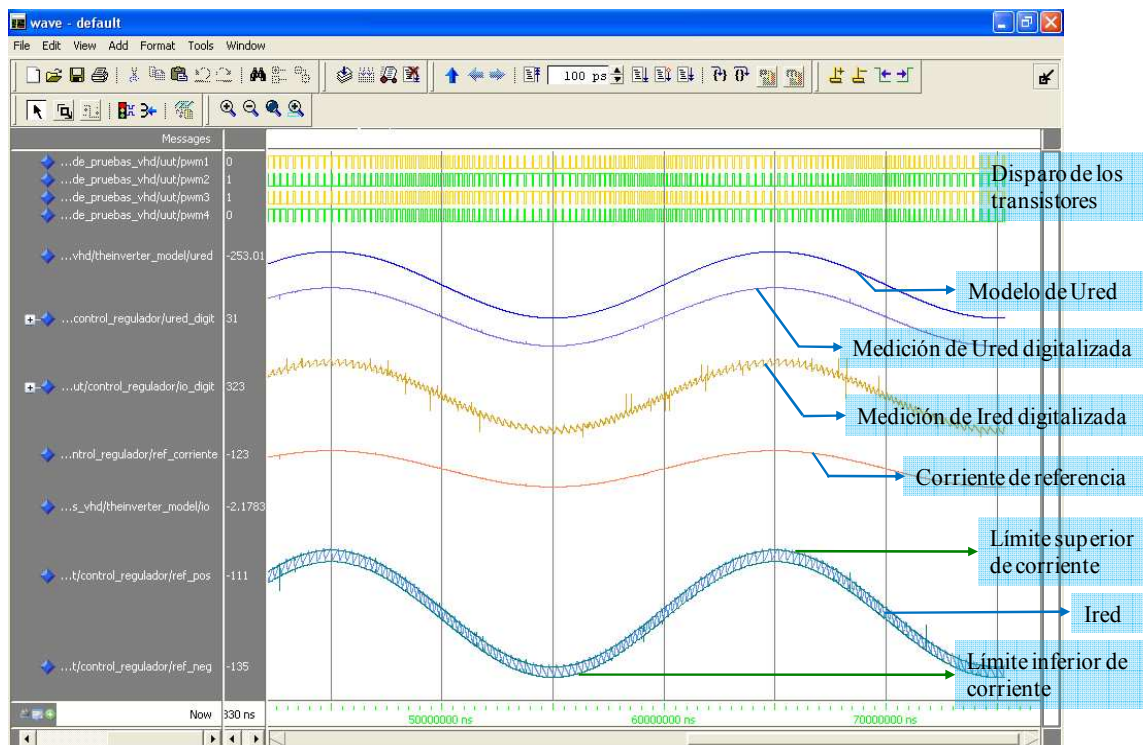


Figura 4.23. Simulación en MODELSIM del inversor con el regulador de corriente por histéresis.

En la simulación de la Figura 4.23 se representan las señales más importantes del sistema de control de corriente. En primer lugar se muestran los disparos de los transistores, que son complementarios dos a dos e incluyen un tiempo muerto de 1 μ s entre conmutaciones. A continuación se representa la señal que emula a la tensión de red ideal, creada a partir del modelo de una senoide. Las siguientes señales referentes a la tensión de red y la corriente de red digitalizadas corresponden en realidad a registros digitales de 10 bits, obtenidos tras la conversión de los ADC's. Ambas señales digitales son tratadas por el sistema de control, quitándoles en cada caso el valor digital correspondiente para que queden centradas en el origen. La señal de tensión es empleada para generar la referencia de corriente, y esta a su vez impone los límites de la banda de histéresis. En esta figura también se puede ver el resultado comentado anteriormente del efecto de la histéresis sobre la corriente de salida del inversor, obteniéndose una señal con forma sinusoidal formada por pequeñas rampas debidas a la carga y descarga de la bobina en serie con la red.



Con esta simulación quedan validados los bloques de control del ADC y del regulador por histéresis.

4.4. VALIDACIÓN EXPERIMENTAL DEL INVERSOR CON LAZO DE CORRIENTE CONECTADO A RED.

Tras haber comprobado mediante simulación el correcto funcionamiento del regulador de corriente por histéresis, se ha implementado el bloque de control en la FPGA y se han realizado tres pruebas fundamentalmente:

- La primera de ellas ha consistido en cortocircuitar la salida del inversor, dejando la bobina de conexión a red como única carga.
- En segundo lugar se ha conectado una carga resistiva a la salida del puente inversor que permita observar el funcionamiento del regulador en carga.
- Finalmente, se ha probado el funcionamiento del inversor conectado directamente a la red.

4.4.1. MONITORIZACIÓN DE SEÑALES DIGITALES

Adicionalmente, se ha diseñado como herramienta de medición, un par de convertidores digital-analógicos (D/A) que permiten visualizar fácilmente el valor de los registros digitales tratados por la FPGA. Estos registros están relacionados con el tratamiento de las medidas de la tensión en la red y la corriente de salida del inversor, que son los únicos parámetros medidos en este inversor. El convertidor D/A, cuyo esquema eléctrico se representa en *Figura 4.24*, está constituido por un generador de pulsos PWM más un filtro paso bajo. Este filtro tiene una frecuencia de corte de 500 Hz, valor suficiente para filtrar las componentes de alta frecuencia de las señales de 50 Hz que se van a medir.

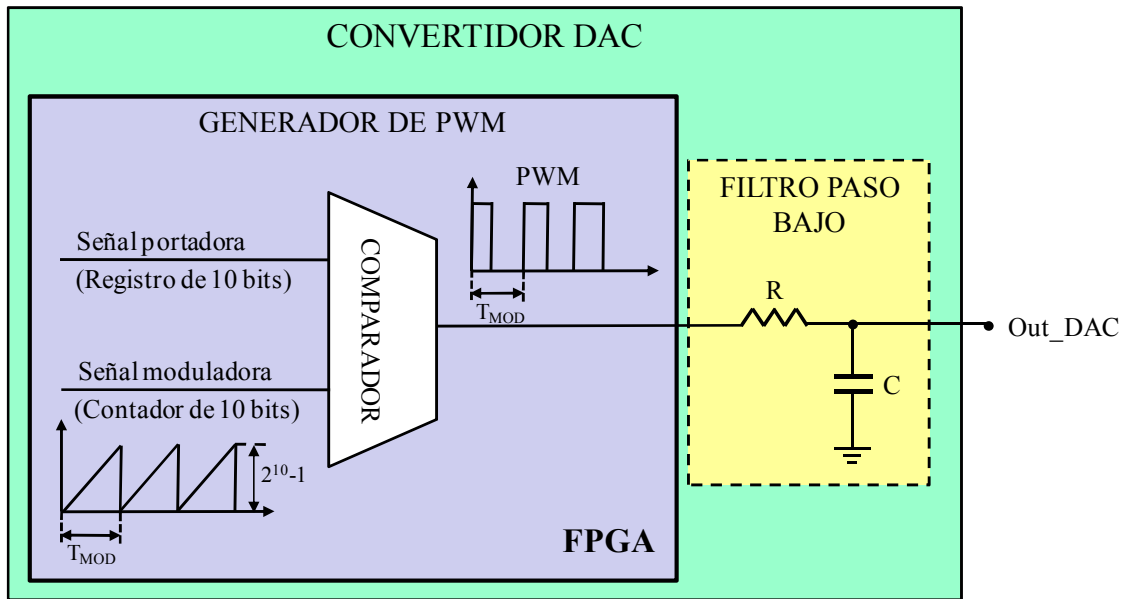


Figura 4.24. Convertidor D/A para la medición de señales digitales de la FPGA.

Para el cálculo de los valores de la resistencia y el condensador del filtro paso bajo, se ha de tener en cuenta que la frecuencia del reloj interno de la FPGA es de 50 MHz, por lo que el paso de integración es de 20 ns. Como la señal moduladora se realiza a través de un contador digital de 10 bits, el periodo de dicha señal moduladora será $T_{MOD} = 20 \text{ ns} \cdot 2^{10} = 20.48 \text{ } \mu\text{s}$. Esto equivale a generar la PWM con una frecuencia de aproximadamente 50 KHz, luego una frecuencia de corte adecuada para el filtro sería de 5 KHz, que corresponde a una década menos. Sin embargo, se calcula el filtro para una frecuencia de corte de 500 Hz, asegurando una atenuación de 40 dB y un desfase despreciable de la señal de red, que funciona a 50 Hz.

Imponiendo un condensador $C = 10 \text{ nF}$ para el filtro paso bajo y sabiendo que la frecuencia de corte del mismo viene dada por la ecuación $F_{CORTE} = 1/(2\pi RC)$, se tiene que el valor de la resistencia del filtro es $R = 31.8 \text{ k}\Omega$.

El esquema de montaje del módulo de conversión de datos D/A consiste en la implementación en la FPGA del hardware digital correspondiente a la generación de la PWM a partir de las señales moduladora y portadora (ver descripción VHDL en ANEXO VIII, modulo_DAC.vhd); y la conexión del filtro paso bajo a los pines libres del puerto VGA, tal como se indica en la Figura 4.25.

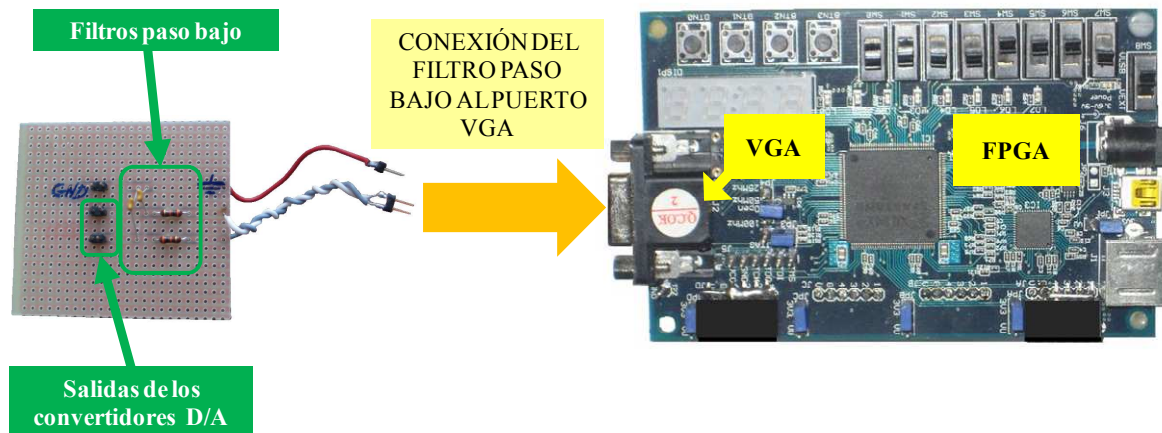


Figura 4.25. Esquema de montaje de los convertidores D/A.

Gracias al convertidor D/A, se pueden visualizar fácilmente los valores digitales tratados por la FPGA, como se puede observar en la Figura 4.26, donde se muestra la tensión de red sensada junto con la salida del convertidor D/A. En este caso, la señal convertida corresponde al registro digital de la tensión de red sensada. Además, para poder representarla con claridad, se le suma al registro un valor de offset que permite obtener valores mayores que 0, ya que el convertidor D/A solo puede tratar señales positivas.

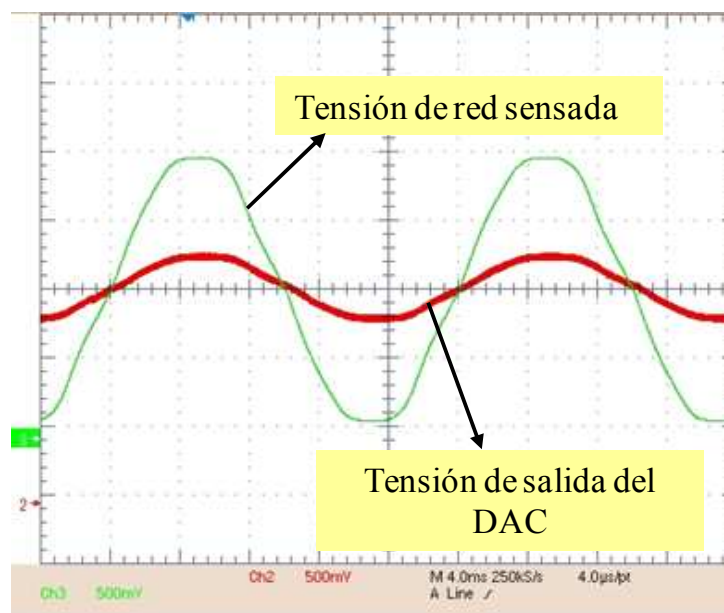


Figura 4.26. Tensión de red sensada y señal de tensión de red digitalizada y pasada por el convertidor D/A.

4.4.2. VALIDACIÓN EXPERIMENTAL DEL REGULADOR POR HISTÉRESIS CON CARGA INDUCTIVA A LA SALIDA DEL INVERSOR

En primera instancia, se ha realizado una prueba experimental del prototipo con el regulador de corriente por histéresis con la bobina de conexión a red (L_2) como única carga (Ver *Figura 4.27*). Esta bobina tiene una inductancia de 8 mH, lo que implica que la impedancia de la misma sea $Z_{L_2} = \omega L = 2\pi fL = 2.51 \Omega$.

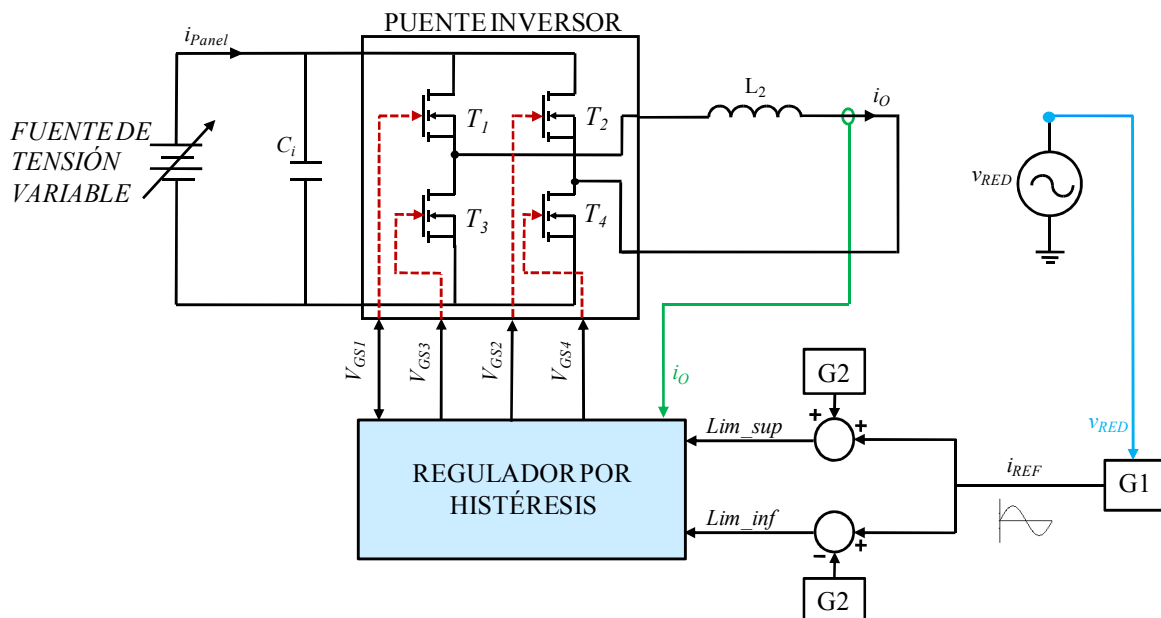


Figura 4.27. Esquema eléctrico empleado para la validación experimental del inversor en cortocircuito funcionando en lazo cerrado.

El objetivo principal de la realización de esta prueba es obtener una corriente de salida que siga a la de referencia empleando pequeñas tensiones de entrada, gracias a la baja impedancia de salida aportada por la bobina.

Para llevar a cabo esta prueba, se ha incluido un selector de parámetros en el hardware digital del inversor, de manera que se permite la elección de 2 diferentes corrientes de referencia, así como 2 posibles límites para la banda de histéresis. Además se ha añadido un par de selectores que permiten ajustar el offset a eliminar de los



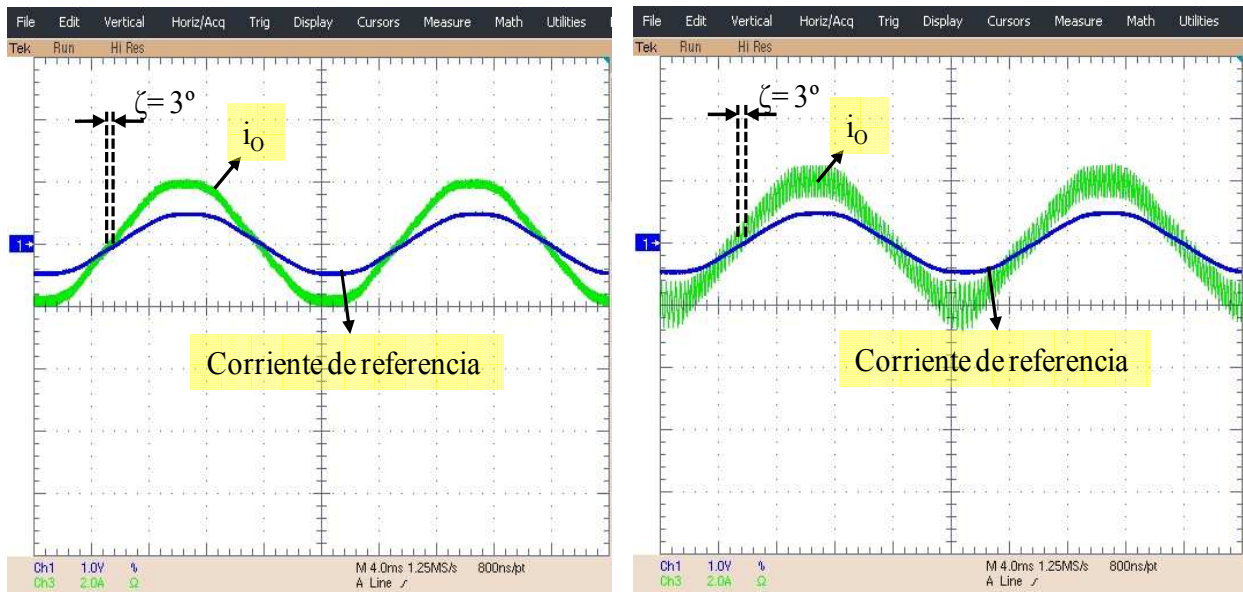
registros digitales a la tensión de red y a la corriente de red, permitiendo ajustar que ambas señales tengan el mismo origen y sean comparables.

Como fuente de tensión variable, se ha emplado la fuente de continua modelo 6015A de HP/AGILENT, que es capaz de dar entre 0 y 500 V, de 0 a 5 A y una potencia total de hasta 1 kW. Ver *Figura 4.7*.

Los principales resultados obtenidos de este ensayo se encuentran en el *ANEXO IX*.

En la *Figura 4.28* se muestra una comparativa entre las 2 bandas de histéresis empleadas para el caso del inversor con la carga inductiva de 8 mH a la salida. Las condiciones de funcionamiento de esta prueba están basadas en una tensión de panel de 100 V, con una referencia de corriente de 2 A de pico. Las señales representadas en ambas figuras son:

- Por una parte, la corriente de referencia, cuya medida se ha obtenido tras la digitalización de la tensión de red sensada y su posterior paso por un convertidor D/A.
- Por otra parte, se muestra en cada caso la corriente de salida del inversor, que está formada por un rizado característico procedente de la carga y descarga de la bobina. Como se puede observar, la corriente de salida de la *Figura 4.28.a* posee un rizado menor que en la *Figura 4.28.b*, lo cual es debido a que en esta última se ha empleado una banda de histéresis más grande. Además, en ambas figuras se puede observar cómo se produce un pequeño retraso de 3° en la corriente de salida respecto de la corriente de referencia. Este desfase tiene su origen en el filtro paso bajo que forma parte del convertidor D/A, cuya frecuencia de corte es de 500 Hz, y que provoca un pequeño retraso de la señal de 50 Hz correspondiente a la corriente de referencia.



a. Banda de histéresis menor.

b. Banda de histéresis mayor.

Figura 4.28. Validación experimental del control del inversor en lazo cerrado con carga inductiva, tensión de panel de 100 V y corriente de referencia de 2 A de pico.

4.4.3. VALIDACIÓN EXPERIMENTAL DEL REGULADOR POR HISTÉRESIS CON CARGA RESISTIVA A LA SALIDA DEL INVERSOR

En segundo lugar, se ha verificado el correcto funcionamiento del inversor con lazo de corriente controlado por histéresis, empleando una carga resistiva a la salida del inversor de valor 106Ω (Ver Figura 4.29).

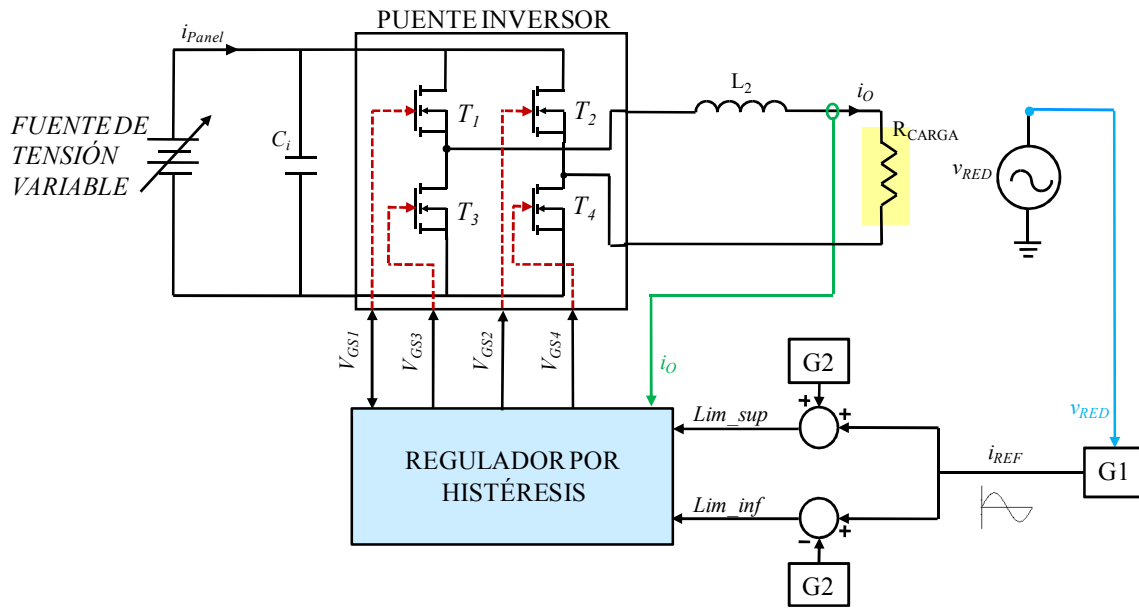


Figura 4.29. Esquema eléctrico empleado para la validación experimental del inversor con carga resistiva de 106Ω funcionando en lazo cerrado.

El objetivo de este ensayo es trabajar en condiciones de funcionamiento más severas que el ensayo de cortocircuito, ya que para alcanzar la misma corriente de referencia en este caso se requieren tensiones de panel más elevadas. Esto se debe a que la impedancia de salida del inversor incrementa de 2.51Ω a 108.51Ω .

Los principales resultados experimentales obtenidos acerca de la realización de este ensayo se pueden observar en el ANEXO IX.

En la Figura 4.30 se representa el resultado obtenido al realizar un ensayo con el prototipo de pruebas en bucle cerrado, empleando una carga resistiva de 106Ω a la salida del inversor. Las condiciones de funcionamiento de este ensayo son las siguientes: Tensión de entrada del puente inversor de 240 V y corriente de referencia de 2 A de pico. En esta captura tomada con el osciloscopio, se puede apreciar que la corriente de salida del inversor sigue a la corriente de referencia, que en este caso se representa como una tensión cuyo origen proviene de la digitalización de la corriente de salida sensada y pasada por un convertidor D/A.

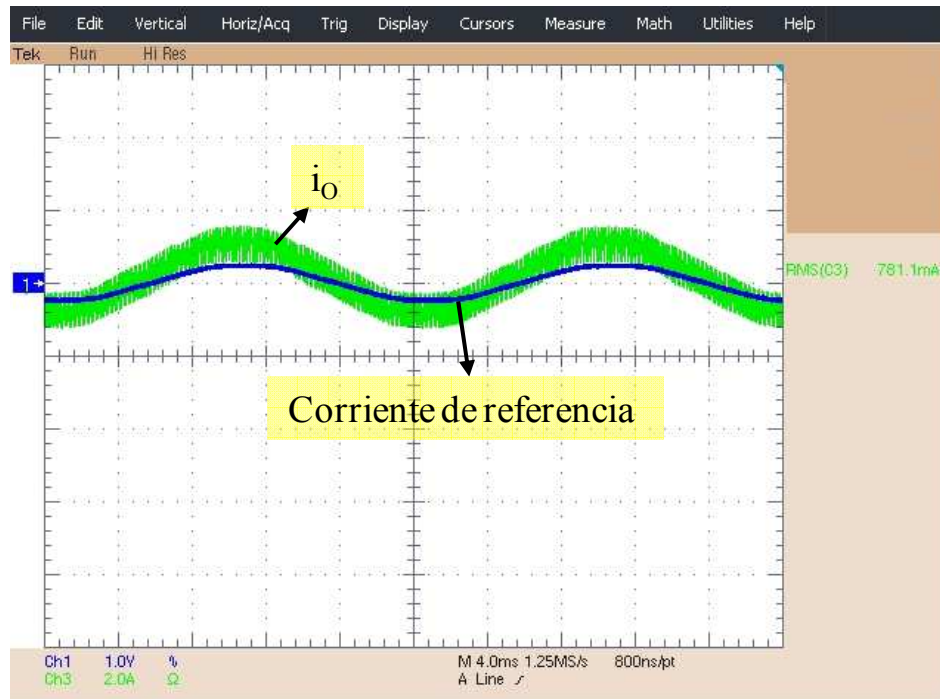


Figura 4.30. Validación experimental del control del inversor en lazo cerrado con carga resistiva ($R=106 \Omega$), tensión de panel de 240 V y corriente de referencia de $2 A_{PICO}$.

En la Figura 4.31 se puede ver una captura en detalle de la tensión de salida del inversor (u_{o_inv}) y la corriente de salida del mismo (i_o) para las condiciones de funcionamiento empleadas en el ensayo de la Figura 4.30. En esta imagen se puede apreciar el rizado de i_o característico del efecto de histéresis, debido a los tramos de carga y descarga de la inductancia L_2 .

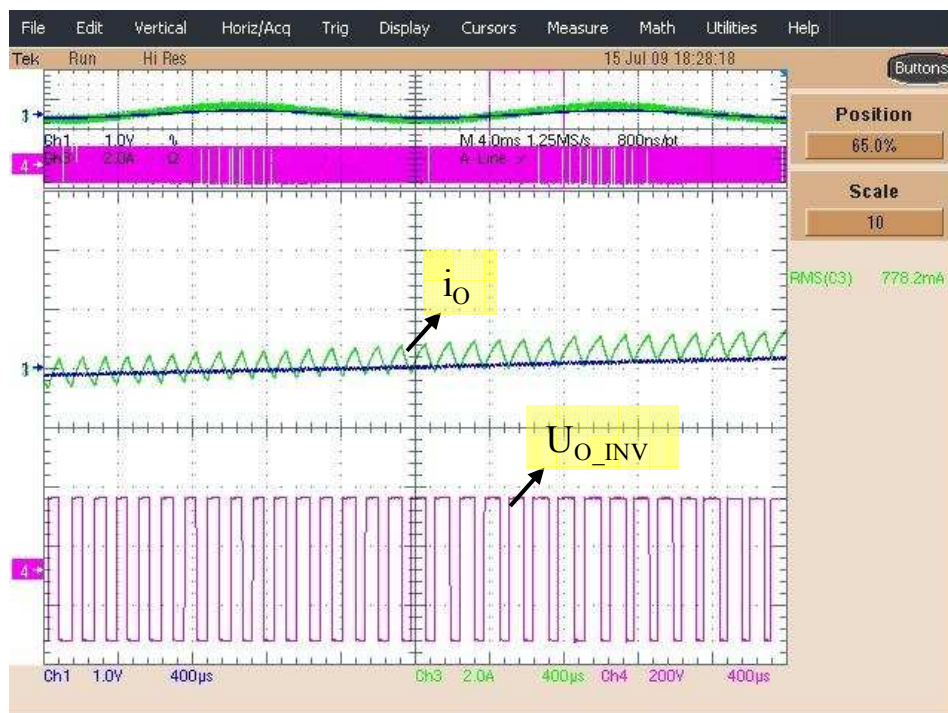


Figura 4.31. Imagen en detalle de la tensión y corriente de salida del inversor para unas condiciones de funcionamiento siguientes: Tensión de panel de 240 V y corriente de referencia de 2 A de pico.

4.4.4. VALIDACIÓN EXPERIMENTAL DEL REGULADOR POR HISTÉRESIS CONECTADO DIRECTAMENTE A LA RED

El último y más determinante paso para la validación experimental del puente inversor con el regulador de corriente por histéresis, es la conexión directa a la red. Ver Figura 4.32.

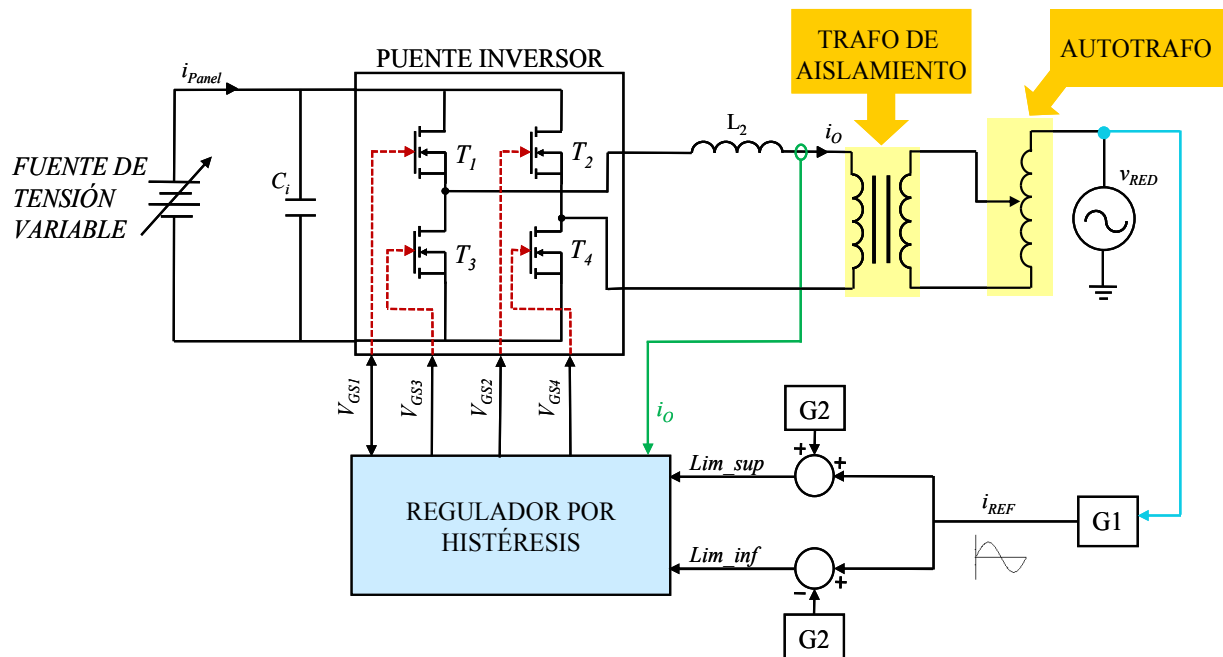


Figura 4.32. Esquema eléctrico empleado para la validación experimental del inversor conectado a red.

Para llevar a cabo este procedimiento, se requieren dos elementos fundamentales:

- Un **autotransformador trifásico**, que permite realizar un incremento regulable de la tensión de la red (Figura 4.33.a), para facilitar las pruebas del prototipo.
- Un **transformador de aislamiento** (Figura 4.33.b), cuya función es la de desacoplar la etapa de potencia de la red eléctrica, imponiendo aislamiento galvánico entre ambos.



a. Autotransformador trifásico.

b. Transformador de aislamiento.

Figura 4.33. Dispositivos empleados para la validación experimental del inversor en lazo cerrado conectado a red.

Los ensayos de conexión a red se han realizado de 2 maneras:

- Se ha llevado a cabo una primera prueba con el esquema eléctrico que se muestra en la Figura 4.32, en la que entre la salida del puente inversor y la etapa de conexión a red solo existe la inductancia de conexión a red L_2 . Ver resultados en ANEXO IX.
- En segundo lugar, se ha sustituido la inductancia L_2 por un filtro en T, tal y como se observa en la Figura 4.34. El objetivo de este dispositivo es reducir el contenido armónico de la corriente inyectada a red. Ver resultados en ANEXO IX.

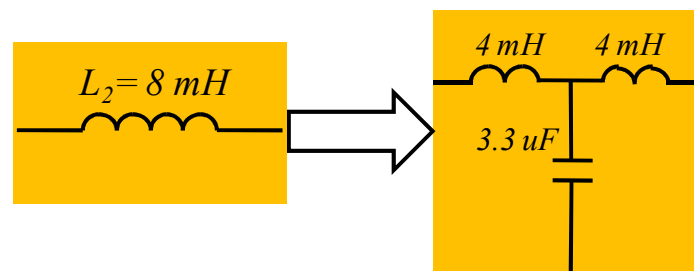


Figura 4.34. Filtro en T empleado para los ensayos del inversor conectado a red.

A continuación se muestra en la Figura 4.35 el resultado de haber realizado un ensayo con el inversor conectado a red para unas condiciones standard de

funcionamiento, como son los 110 V eficaces característicos de algunas redes eléctricas como la americana, o equivalente a la red europea considerando un transformador de acoplamiento a la red con relación 1:2. Las condiciones de funcionamiento para la realización de esta prueba son las siguientes: Tensión de entrada de 186 V, referencia de corriente de 2 A de pico y filtro LCL a la salida del inversor.

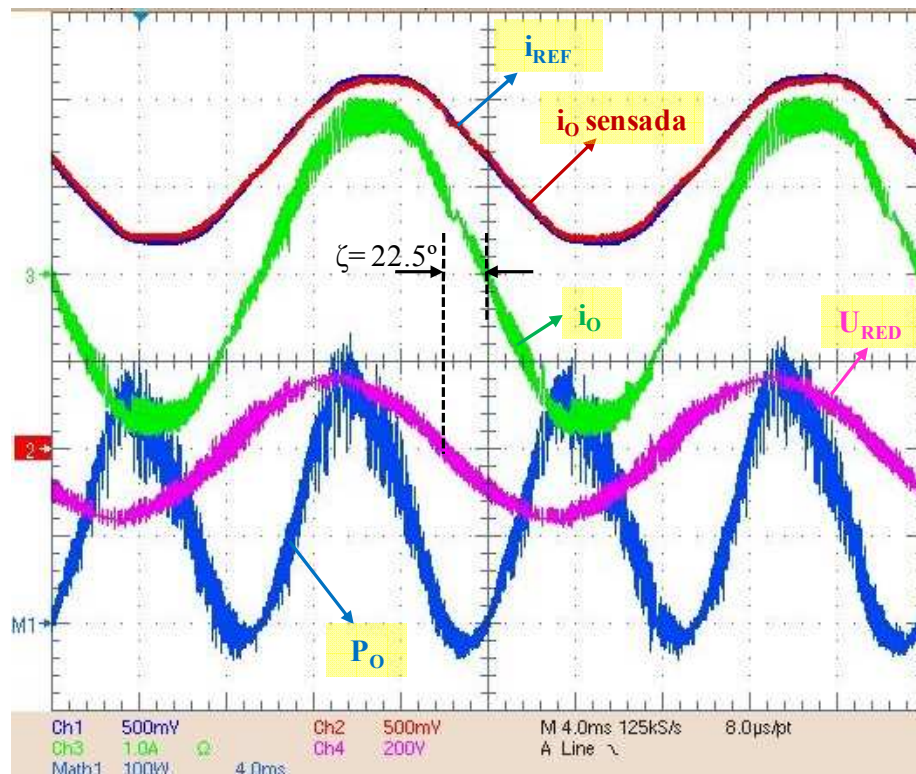


Figura 4.35. Resultado del ensayo referente al lazo de corriente del inversor con filtro LCL. Condiciones de funcionamiento: $U_{RED}=110 V_{ef}$, $I_{REF}=2 A_{pico}$.

En la Figura 4.35 se puede observar que la corriente de salida (i_o) sigue con bastante precisión a la corriente de referencia (i_{REF}). Por otra parte, se aprecia que la tensión de red (u_{RED}) está desfasada 22.5° con respecto a i_o . Este retraso se debe a que, mientras que i_o se genera a partir del sensado de la tensión de red obtenida directamente de la red², la tensión aquí representada como u_{RED} es la tensión medida en el

² La tensión a partir de la que se genera i_o se mide directamente en la red sin pasar por el transformador de aislamiento y el autotransformador porque en las condiciones de las pruebas del prototipo la tensión de red vista por el inversor tiene una amplitud variable, lo que no es compatible con el algoritmo de generación de la corriente de referencia. En funcionamiento en condiciones normales con tensión de red de amplitud fija, la tensión para generar la corriente de referencia sería la tensión de red vista por el inversor y este desfase no se produciría.



transformador de aislamiento, que a su vez está conectado al autotransformador y éste a su vez a la red.

Finalmente, en la *Figura 4.35* se representa una forma de onda que muestra la potencia de salida instantánea del inversor (P_O), obtenida como el producto de i_O y u_{RED} . Esta forma de onda calculada con el osciloscopio no es de forma rigurosa la potencia real inyectada a la red, ya que i_O no es exactamente la corriente inyectada en la red, sino que es la corriente de salida del inversor, que se diferencia de la corriente inyectada a la red, a través del transformador de aislamiento, en la corriente que circula por el condensador del filtro LCL. Sin embargo, esta medida nos da una idea aproximada de la potencia inyectada en la red. Atendiendo a esta consideración, la potencia media entregada por el inversor en este ensayo, es de 124.3 W, por lo que, teniendo en cuenta que la fuente de continua entrega 152.52 W, el rendimiento de este sistema sería de aproximadamente 81.5 %. Teniendo en cuenta que debido al desfase entre la corriente inyectada a la red y la tensión de red vista por el inversor (tensión en el transformador de aislamiento) se está inyectando una cierta cantidad de potencia reactiva, el rendimiento del inversor sería previsiblemente mejor si la corriente inyectada a la red y la tensión de red vista por el inversor estuvieran en fase.

Las medidas realizadas con el inversor en lazo cerrado están recogidas en el *ANEXO IX*, donde se incluye una tabla con los principales resultados obtenidos. De estos resultados hay que destacar que se han llegado a emplear tensiones de entrada del lado de continua de hasta 315 V, de tensión de red 210 V_{ef} y se han llegado a inyectar hasta 200 W en la red. Los rendimientos obtenidos en estas pruebas se encuentran entre el 70% y más de 85%, dependiendo en cada caso de la referencia de corriente y el uso del condensador interpuesto entre las 2 bobinas de 4 mH (filtro LCL) o no. En general se obtienen mejores rendimientos con el filtro LCL, lo que puede deberse a que cuando se utiliza el condesador, solo por una de las bobinas del filtro circulan los armónicos de corriente de alta frecuencia y por lo tanto las pérdidas asociadas al incremento de resistencia por el efecto skin o pelicular solo se producen en una de las 2 bobinas.



4.5. CONCLUSIONES

El resultado más importante de este capítulo es la validación experimental del inversor conectado a la red eléctrica inyectando potencia. Además, se ha caracterizado el inversor en lazo abierto y en lazo cerrado (sincronizado con la red) sin conexión a la red y con carga inductivo resistiva.

Para llegar a estos resultados ha sido necesario el diseño del control digital que gobierna el inversor y del control de los convertidores A/D que traducen a valores digitales las señales de los sensores.

El control del inversor en lazo abierto se ha basado en la generación de una forma de onda PWM a partir de tablas que contienen los ángulos de corte previamente calculados con un pequeño programa en Matlab. El hardware digital es muy simple y flexible.

El control digital del inversor en lazo cerrado se ha centrado en el lazo de corriente, que asegura que la corriente inyectada a la red tiene la misma forma de onda que la tensión de red y está sincronizada con ella. Para ello se ha utilizado un regulador por histéresis, obteniendo unos buenos resultados en cuanto a forma de onda de la corriente inyectada a la red.

El control de los convertidores A/D también ha sido presentado en este capítulo y validado mediante pruebas experimentales.



CAPÍTULO 5.

PROPUESTA DE ALGORITMO DE MPPT SIN SENSORES DEL LADO DE CC.



En este capítulo se describe el algoritmo propuesto para el control del seguimiento del punto de máxima potencia de los paneles fotovoltaicos. Esta descripción se acompaña de una simulación en PSIM.

5.1. EL ALGORITMO DE CONTROL PARA EL SEGUIMIENTO DEL PUNTO DE MÁXIMA POTENCIA (MPPT).

En todo sistema basado en paneles solares es muy importante extraer la máxima potencia de los mismos, ya que de este modo se logra un funcionamiento más eficiente del mismo. En este aspecto, existen muchos algoritmos destinados a conseguir obtener la máxima potencia del inversor y, por regla general, estos algoritmos se centran en la medición de los parámetros de entrada del puente inversor, como son la tensión de entrada y la corriente entregada por los paneles solares. Sin embargo, en este proyecto el objetivo es conseguir un sistema de control lo más barato posible. Por ello, es importante medir el menor número de magnitudes posibles, ya que de este modo se disminuye la cantidad de sensores y la electrónica de instrumentación necesaria.

De acuerdo con estas consideraciones de partida, en el proyecto UNISOL se ha incluido un algoritmo de control MPPT que solo requiere la medición de los parámetros conocidos de antemano, que son la tensión de la red y la corriente a la salida del puente inversor.

El bloque de control del MPPT se muestra en la *Figura 5.1*. El hardware digital de este sistema de control se centra en el del regulador de corriente por histéresis, con la principal modificación del MPPT en lugar de una constante. De este modo, dicho bloque del MPPT se encarga de generar la corriente de referencia a partir de la tensión de red y del disparo de uno de los transistores. Esta referencia se caracteriza por buscar el funcionamiento del sistema en el punto de máxima potencia del panel fotovoltaico.

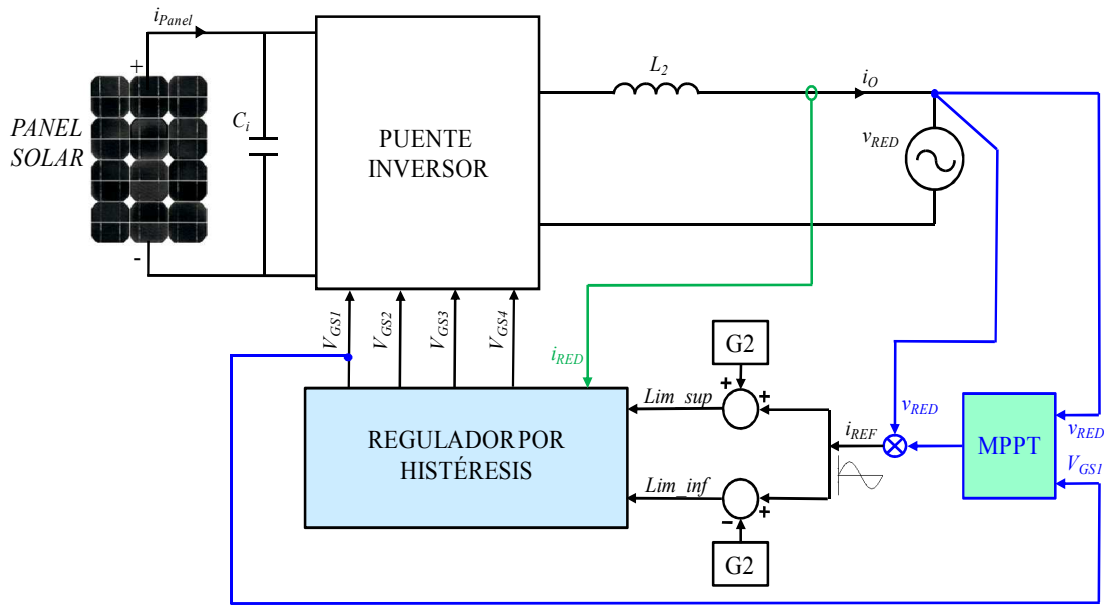


Figura 5.1. Diagrama de bloques del inversor en lazo cerrado con control MPPT.

Para llevar a cabo el seguimiento del punto de máxima potencia, es necesario conocer el funcionamiento del panel solar fotovoltaico. En este aspecto, una forma común de representar el funcionamiento del panel es a través de las curvas características de tensión y potencia frente a la corriente generada. Como se puede observar en la Figura 5.2, cuando el panel está en circuito abierto, la corriente entregada por el mismo es nula, aumentando lentamente a medida que la corriente generada aumenta. Este incremento corresponde con un aumento de potencia entregada por el panel, hasta que se llega a un punto en el que la potencia es máxima, que es el denominado Punto de Máxima Potencia (PMP). A partir de este punto, tanto la tensión del panel como la potencia del inversor decrecen bruscamente con la corriente, llegando a un punto en el que el panel no puede dar más energía.

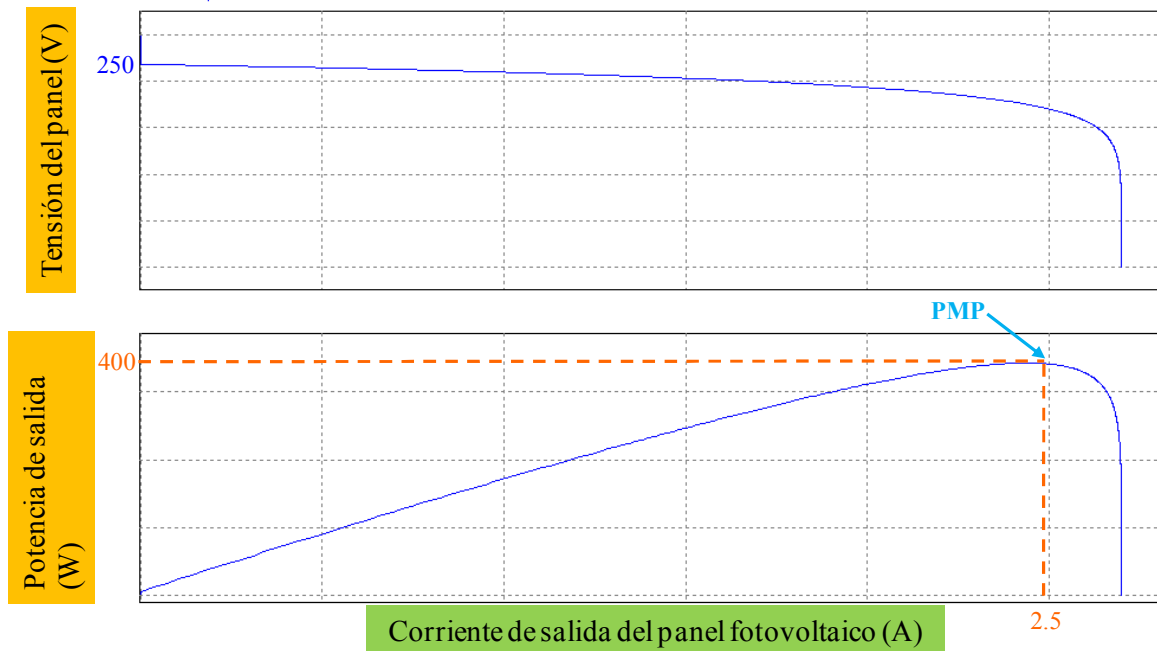


Figura 5.2. Curvas de corriente y potencia frente a tensión en un panel solar fotovoltaico.

Como ya se ha mencionado, el control del MPPT utiliza las siguientes señales:

- Referencia de corriente: es la señal de salida de este bloque. Se calcula multiplicando la tensión de red por la amplitud de la corriente de referencia, que es la verdadera variable de control calculada por el lazo.
- Tensión de red: es una entrada a este bloque. Sirve para generar la referencia de corriente en fase con la red y sincronizada con ella.
- Pulsos de disparo de uno de los transistores: es una señal de entrada. A partir de esta señal se evalúa el ciclo de trabajo medio y si éste aumenta o disminuye, lo que determina si la amplitud de la referencia de corriente aumenta o disminuye.

El condensador de entrada del panel fotovoltaico juega un papel muy importante en este aspecto, ya que el ancho de los pulsos de los transistores depende de la energía entregada por este condensador en cada instante. A medida que el inversor demanda más corriente al panel, la tensión en el condensador de entrada cae y los transistores se activan durante más tiempo (mayor ciclo de trabajo) para seguir atendiendo a las demandas de energía.



El funcionamiento de este lazo de control se basa en estimar en qué condiciones la potencia que se pide al panel es mayor de la que puede dar. La amplitud de la corriente inyectada a la red (y, por tanto, la referencia de corriente) determina la potencia inyectada a la red. Supóngase que se trabaja en un punto estable (el panel pueda dar la potencia inyectada a la red) y se aumenta la referencia de corriente, es decir, la demanda de potencia al panel. Si la referencia de corriente es muy grande y la potencia inyectada a la red es mayor de la que puede dar el panel, el condensador de entrada del inversor se descargará, aportando la diferencia entre la potencia entregada por el panel y la inyectada a la red. Si esta situación se prolonga durante mucho tiempo, el condensador de entrada se irá descargando paulatinamente. Dado que el lazo de corriente intenta inyectar en esta situación una amplitud de corriente constante, al disminuir la tensión del condensador, el ciclo de trabajo medio de los transistores del inversor, aumentará. Este es el hecho clave para el funcionamiento de este lazo. **La medida del ciclo de trabajo medio a lo largo de un semiciclo de red se toma como estimador para determinar si el panel puede dar en régimen permanente toda la potencia que se quiere inyectar en la red o no:**

- Si el ciclo de trabajo medio durante un semiciclo aumenta indefinidamente, es porque se debe compensar el descenso de la tensión en el condensador de entrada del inversor. Por tanto, la potencia inyectada en la red es excesiva y se debe bajar la referencia de corriente.
- Si el ciclo de trabajo medio durante un semiciclo se mantiene constante es porque existe un equilibrio entre la potencia que proporciona el panel y la inyectada a la red, de manera que la tensión del condensador de entrada se mantiene constante.

El algoritmo de control del MPPT se inicia en un punto de funcionamiento del panel en el que entrega poca corriente y va demandando más corriente al panel de forma discreta, espaciando estas peticiones el tiempo suficiente para que el condensador de entrada llegue a su régimen permanente de funcionamiento. Cada vez que se pide un poco más de corriente al panel, la tensión en el condensador de entrada decae, haciendo que el ciclo de trabajo medio se incremente en cada semiciclo de red para poder satisfacer los requerimientos energéticos. El MPPT mide estas variaciones del ciclo de

trabajo medio en cada semiciclo positivo de red y, en caso de percibir que estos incrementos se dan durante un periodo prolongado, actúa sobre la corriente de referencia, obligando a que se demande menos corriente del panel. Esto ocurre cuando se sobrepasa el punto de máxima potencia, ya que a partir de aquí, la energía que entrega el panel sufre un descenso abrupto y el condensador de entrada se descarga durante más tiempo, que es lo que detecta el sistema de control.

En la *Figura 5.3* se ilustra el flujograma correspondiente al funcionamiento que se acaba de describir del lazo de control del MPPT.

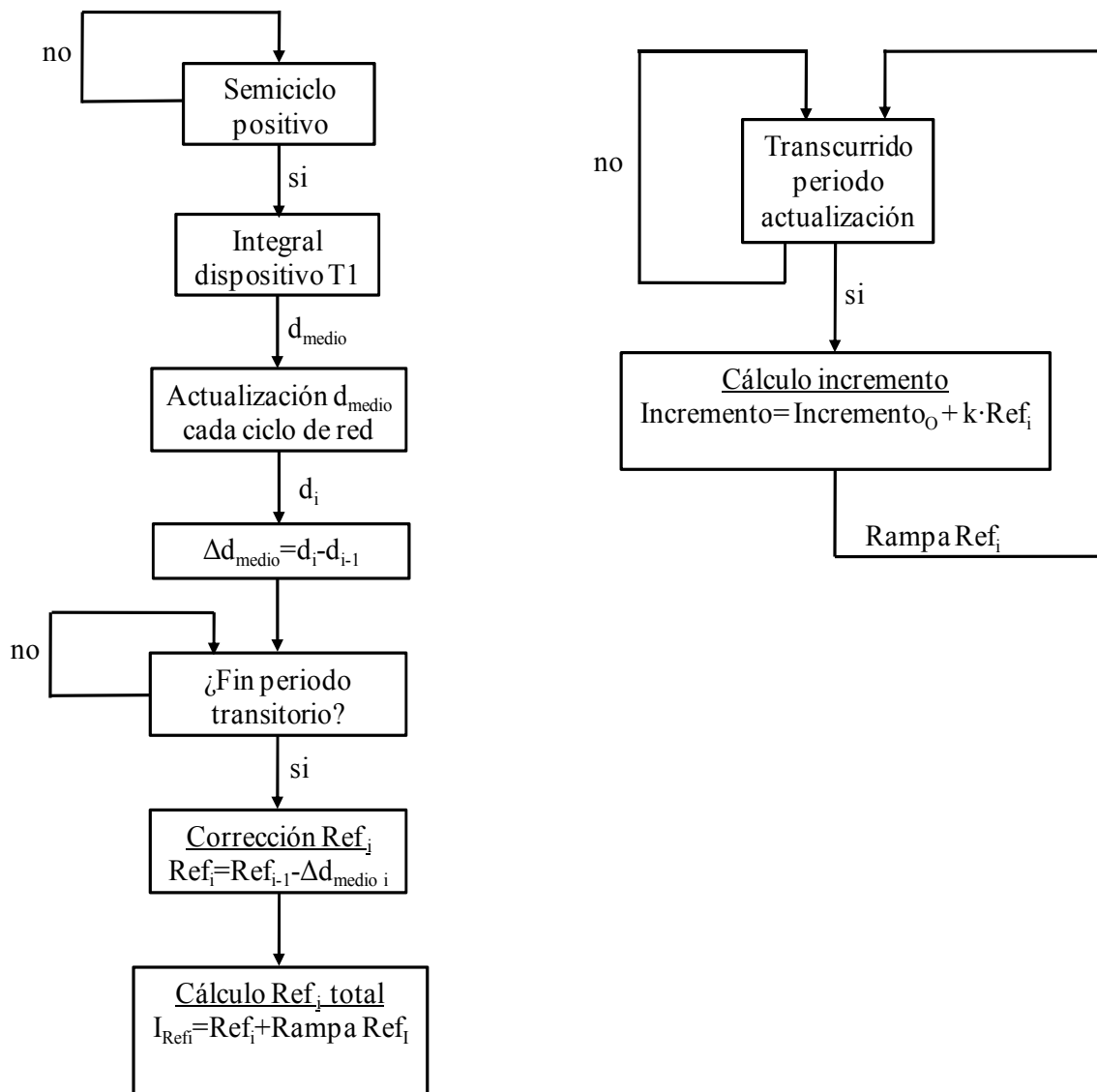


Figura 5.3. Flujograma correspondiente al control del MPPT.



La característica principal de este lazo es que busca el punto de máxima potencia a través de la observación de la tensión en el condensador de entrada cada cierto periodo. Por este motivo, se trata de un sistema muy lento, ya que la inercia del condensador de entrada es muy grande debido a su gran tamaño.

5.2. SIMULACIÓN DEL INVERSOR CON EL LAZO DE CORRIENTE Y EL CONTROL DEL MPPT PROPUESTO

Para llevar a cabo el seguimiento del punto de máxima potencia, se ha empleado la herramienta de simulación de circuitos electrónicos PSIM. La elaboración de este bloque de control se ha realizado empleando un modelo de panel fotovoltaico capaz de entregar 400 W, que es la potencia máxima que puede aportar el inversor según las pruebas realizadas en lazo abierto (Ver *ANEXO X*). Este modelo ha sido diseñado por el Grupo de Sistemas Electrónicos de Potencia del Departamento de Tecnología Electrónica para proyectos basados en energía fotovoltaica.

El circuito electrónico diseñado para las simulaciones del sistema de control completo (lazo de corriente + MPPT) se representa en la *Figura 5.4*.

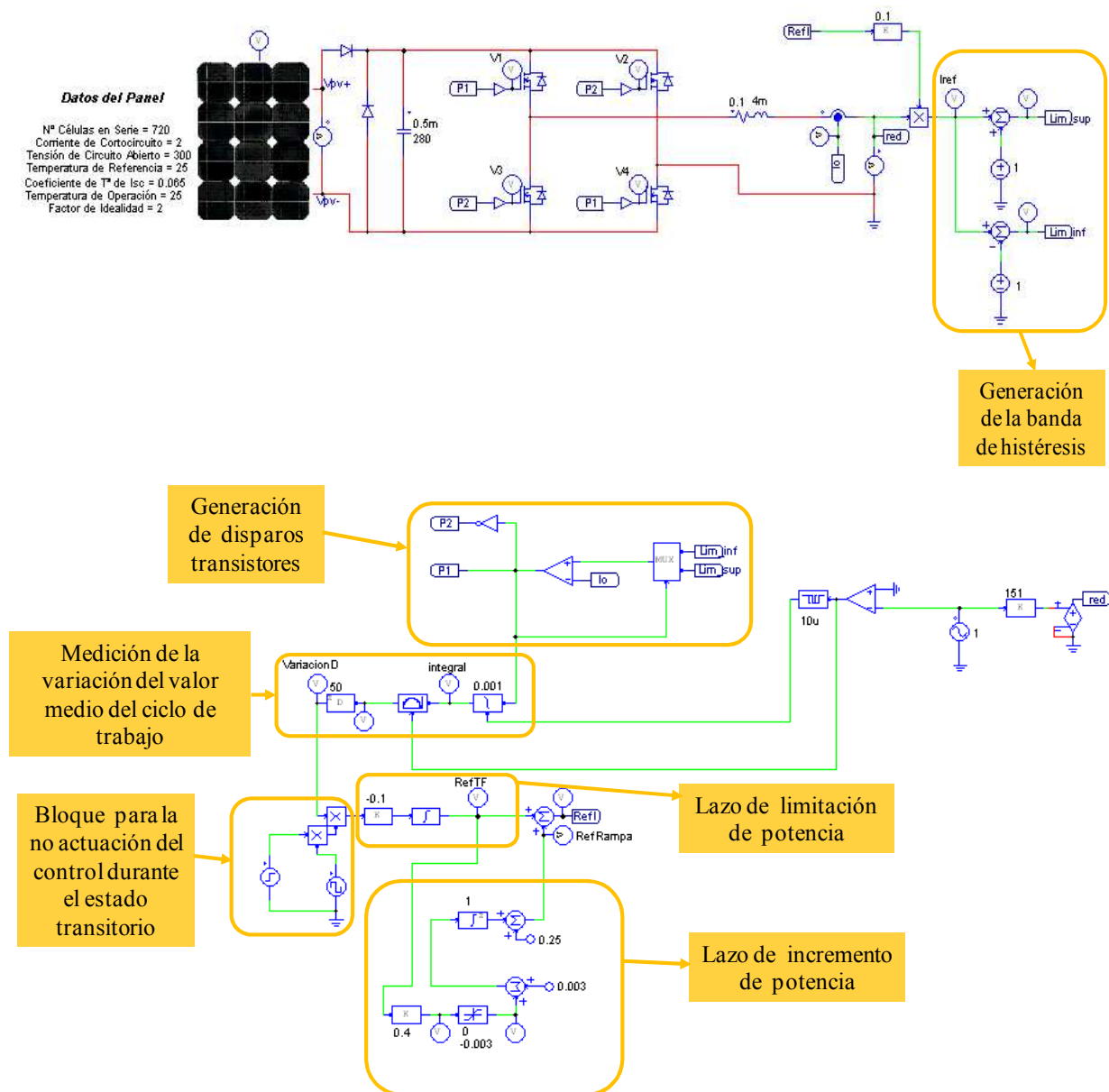


Figura 5.4. Esquemático empleado para la simulación en PSIM del control del inversor en lazo cerrado con MPPT.

El esquema del algoritmo control del MPPT propuesto para este inversor se puede subdividir en los bloques que se describen a continuación:

- Un bloque que mide la variación del valor del ciclo de trabajo medio de los pulsos de disparo en cada semiciclo positivo de red. En concreto, se basa en el pulso de disparo generado para el disparo de los transistores 1 y 4, que son iguales.



- Un bloque que genera unos periodos de tiempo durante los cuales el sistema de control del MPPT no actúa. De este modo, no se tienen en cuenta los tiempos en los cuales el condensador de entrada inyecta corriente instantánea para satisfacer los requerimientos producidos por el lazo de incremento de potencia. Este tiempo de transitorio es un parámetro ajustable que dependerá de las necesidades del sistema.
- El lazo de incremento de potencia es el encargado de aumentar en cantidades discretas los requerimientos de corriente que se demanda de los paneles fotovoltaicos. Este lazo está activo hasta que los paneles sobrepasan el punto de máxima potencia.
- El lazo de limitación de potencia. Este módulo se encarga de corregir la corriente de referencia cuando se sobrepasa el punto de máxima potencia, haciendo que los paneles lleguen a trabajar en un punto de funcionamiento muy cercano al punto de máxima potencia.

En la *Figura 5.5* se representan los principales resultados de la simulación del inversor monofásico con el control integrado del MPPT. En esta simulación se observa con claridad el incremento de la potencia entregada por el panel fotovoltaico durante el periodo en el que actúa el lazo de incremento de potencia. Esto se debe a que la potencia demandada del panel no ha llegado a su punto máximo. Durante este periodo de tiempo, el lazo de limitación de potencia no actúa, hasta que llega un momento en el que el panel no es capaz de dar la energía que se le pide, en el cual comienza a actuar el lazo de limitación de potencia. A partir de aquí, el lazo de incremento de potencia permanece inactivo, mientras el otro lazo corrige el error, llegando a un punto de funcionamiento muy próximo al punto de máxima potencia. En esta representación se puede observar también cómo se va descargando el condensador de entrada (C_i) de forma aproximadamente lineal hasta que se sobrepasa el punto de máxima potencia, instante en el cual la caída de tensión es mucho más pronunciada. Este efecto es detectado por el lazo de limitación de potencia, que comienza a actuar haciendo que el condensador vuelva a cargarse hasta un determinado valor, correspondiente a la tensión del panel con la que éste aporta la máxima potencia.

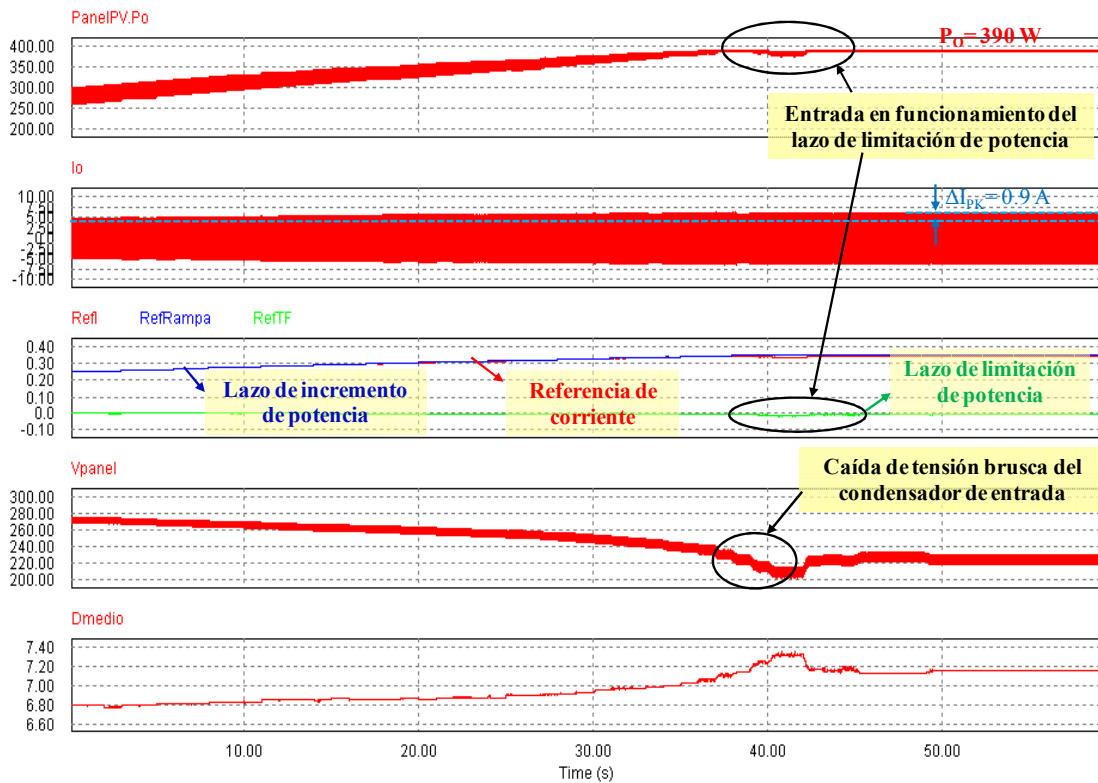


Figura 5.5. Resultado de la simulación en PSIM del control del MPPT.

5.3. CONCLUSIONES

En este capítulo se ha propuesto un algoritmo de seguimiento del punto de máxima potencia (MPPT) para un inversor monofásico con conexión a red basado en la medida de la tensión de red y en el ciclo de trabajo medio, sin necesidad de sensores en el lado de corriente continua del inversor.

La **principal ventaja** del mecanismo propuesto es que no necesita sensores adicionales del lado de continua, lo que repercute en un menor coste, menor mantenimiento, mayor robustez y simplicidad del sistema.

Uno de los mayores **inconvenientes** del algoritmo propuesto es la lentitud de actuación. Como se ha explicado, el lazo determina si se puede pedir más potencia al panel a través de la evolución de la tensión en el condensador de entrada del inversor. Esta evolución es normalmente muy lenta porque la capacidad de este condensador suele ser bastante elevada y porque el desequilibrio entre potencia inyectada a la red y



potencia proporcionada por el panel es relativamente pequeño y el condensador se descarga lentamente. Además, para obviar el efecto de la transición de un punto a otro de la característica V-I del panel y el cambio en la tensión en el condensador debido a ello y no al desequilibrio de potencia, el lazo del MPPT deja de actuar un tiempo después de cada cambio de la referencia de corriente, lo que hace que el lazo se aún más lento.

El algoritmo propuesto ha sido validado mediante simulación con PSIM. La implementación de este algoritmo se encuentra en una etapa preliminar y forma parte de los trabajos futuros.



CAPÍTULO 6.

ESTUDIO ECONÓMICO

En el presente capítulo se aborda la estimación de los costes derivados de la realización del proyecto. Los costes totales del proyecto comprenden por una parte los costes de desarrollo de ingeniería y documentación; y por otra parte se estima un 10 % más de los costes totales para otros gastos. No se tienen en cuenta en este análisis los costes derivados del resto de tareas realizadas en el proyecto UNISOL correspondientes a la elaboración del filtro LCL y el filtro de EMI.

6.1. COSTES DE DESARROLLO

Los costes de desarrollo engloban el análisis, diseño, montaje y documentación de las distintas fases que forman parte del desarrollo de este proyecto. Para ello se estima un coste de 50 €/h en conceptos de trabajo de ingeniería y de 20 €/h en conceptos de documentación.

ACTIVIDAD	€/hora	nº horas	COSTE
Desarrollo de Ingeniería	50	1232	61600 €
Diseño y montaje del hardware correspondiente al inversor monofásico del proyecto UNISOL	50	356	17800 €
Análisis y diseño del hardware digital implementado en la FPGA para el control del inversor	50	876	43800 €
Documentación	20	80	1600 €
Total			63200 €



6.2 COSTES DE LOS MATERIALES

Los gastos de material hacen referencia a la compra de los sensores, fuentes auxiliares, la FPGA y otros componentes electrónicos que han sido necesarios para el desarrollo y la puesta en marcha del prototipo de pruebas correspondiente al inversor monofásico.

Materiales	Cantidad	Coste/Ud.	COSTE
Transductor LV25-P	2	66,06 €	132,12 €
Transductor LTS 25-NP	3	16,37 €	49,11 €
MOSFET IRF740	16	4,05 €	64,80 €
Driver IR2110	10	4,75 €	47,50 €
Tarjeta de evaluación de FPGA, modelo Basys	3	42,59 €	127,77 €
Convertidor ADS7887	5	39,6 €	198,00 €
Convertidor CA/CC, modelo TOM 12215	2	39,47 €	78,94 €
Convertidor CC/CC, modelo TEL 5-1211	1	34,09 €	34,09 €
Otros (Disipador, PCB's, cables, componentes electrónicos)	-	-	63,63 €
Total			795,96 €

6.3. COSTE TOTAL DEL PROYECTO

Finalmente, en el coste total del proyecto se incluyen otros gastos correspondientes a la amortización del material de laboratorio empleado para la

realización de las pruebas experimentales, así como a la amortización de las licencias necesarias para el software empleado. Estos gastos se incluyen como un porcentaje del coste total, siendo este porcentaje del 7%.

ACTIVIDAD	COSTE
Desarrollo de Ingeniería y Documentación	63200 €
Materiales	795,96 €
Otros gastos (7%)	4479,72 €
Total	68475,68 €

6.4. PRESUPUESTO DEL PROYECTO

Una vez evaluados los costes totales del proyecto, el presupuesto total del proyecto es el siguiente:

TIPO DE COSTE	COSTE
Coste de desarrollo	68475,68 €
I.V.A. (16%)	10956,11 €
Total	79431,79 €

El coste total del proyecto asciende a setenta y nueve mil ciento sesenta y nueve euros y cincuenta y ocho céntimos, con I.V.A. incluido.



CAPÍTULO 7.

CONCLUSIONES Y TRABAJO FUTURO



7.1. CONCLUSIONES.

Este proyecto fin de carrera se ha enmarcado dentro del proyecto UNISOL, que ha consistido en el diseño y construcción del prototipo de un inversor de etapa única para la generación distribuida de la energía procedente de un generador fotovoltaico, con el objetivo principal de lograr un sistema de conversión de energía de bajo coste, reducidas dimensiones, modularidad y escalabilidad.

El presente proyecto fin de carrera se ha circunscrito al diseño, construcción y validación del inversor, tanto de la etapa de potencia del inversor como de su control en lazo abierto como conectado a la red eléctrica. Esta ha sido una de las primeras experiencias en el Grupo de Sistemas Electrónicos de Potencia con un inversor monofásico conectado a red.

Las principales valoraciones que se han extraído de la realización de este proyecto son las siguientes:

- Se ha logrado conectar a la red eléctrica un inversor monofásico con control digital mediante una FPGA.
- El sistema diseñado y construido se puede dividir en los siguientes bloques:
 1. etapa de potencia del inversor monofásico, que consiste en un puente completo de transistores MOSFET con sus correspondientes circuitos de disparo y disipador térmico.
 2. etapa de instrumentación y acondicionamiento que contiene los sensores y conversores analógico-digitales (5 canales basados en convertidor A/D serie) para la medición de las magnitudes eléctricas necesarias para realizar el control en lazo cerrado.
 3. etapa de control digital, basada en una tarjeta comercial con FPGA que implementa los algoritmos de control para la generación de los disparos de los transistores del puente inversor.
 4. sistema auxiliar de alimentación para los sensores, drivers y etapa de control, basadas en convertidores comerciales.

- Se ha validado experimentalmente el funcionamiento de las diversas PCB's diseñadas en este proyecto, que son:
 1. La tarjeta de la etapa de potencia, sensores y fuentes de alimentación auxiliares.
 2. La tarjeta de acondicionamiento y conversión de datos A/D.
- Se ha logrado hacer funcionar el inversor en lazo abierto como inversor autónomo con carga resistiva. Este modo de funcionamiento ha servido para caracterizar los límites de tensión y potencia del inversor. La máxima potencia que se ha extraído del inversor es, aproximadamente, 400 W. Teóricamente la potencia que puede manejar el inversor es mayor, pero en el montaje práctico se ha limitado al valor anterior, debido a la conducción cruzada de los MOSFET de la misma rama. Esta anomalía ha aparecido en la práctica por la limitación de los circuitos de disparo para contrarrestar el efecto Miller de los MOSFET.
- El inversor conectado a red produce una corriente con una forma de onda similar a la tensión de red, gracias al lazo de control de corriente basado en un control por histéresis implementado en una FPGA. La conexión a red se ha realizado a través de un autotransformador. La tensión de red ha llegado hasta 220 V_{ef}.
- El lazo de control de corriente basado en el regulador de histéresis obtiene tanto mejores resultados en cuanto similitud de las formas de onda de corriente y tensión de red cuanto menor es la tensión de red. Esto valida la estrategia de control y sugiere que una optimización y rediseño de la etapa de potencia e instrumentación podría proporcionar mejores resultados.
- El acoplamiento del inversor a la red se ha realizado a través de una inductancia, que en algunos ensayos se ha separado en dos inductancias serie con un condensador formando un filtro LCL. Los armónicos de corriente inyectados a la red son mucho menores.
- Se ha propuesto un algoritmo de seguimiento del punto de máxima potencia sin necesidad de medir la tensión y corriente del lado de continua, basándose



únicamente en la medida de la corriente inyectada y de la tensión de red. Este algoritmo se ha validado mediante simulación.

7.2. TRABAJOS FUTUROS.

Entre las mejoras a introducir en este proyecto se destacan las siguientes:

- Rediseñar la PCB de potencia para mejorar los circuitos de disparo.
- Optimizar el filtro de conexión a red y evaluar el contenido armónico de la corriente inyectada.
- Optimizar los parámetros del control digital del lazo de corriente (banda de histéresis en función de la amplitud).
- Probar experimentalmente el algoritmo de control del MPPT propuesto.



BIBLIOGRAFÍA Y **REFERENCIAS.**



Referencias bibliográficas

Daniel W. Hart, “Electrónica de Potencia”, Ed. Prentice Hall, 2001.

Douglas J. Smith, “HDL Chip design”, Ed. Doone Publications, 1996.

Documentos

Informe final proyecto UNISOL (CP06: Inversor de Etapa Unica con MPPT para conexión a Red de Sistemas Fotovoltaicos)

Documentos web

“Digilent PmodAD1™ Analog To Digital Module Converter Board Reference Manual”, disponible en la dirección web

http://www.digilentinc.com/Data/Products/PMOD-AD1/Pmod%20AD1_rm.pdf

“Digilent PmodAD1™ Analog To Digital Module Converter Board Schematics”, disponible en la dirección web

http://www.digilentinc.com/Data/Products/PMOD-AD1/PmodAD1_sch.pdf

“Digilent Basys Board Reference manual“, disponible en la dirección web

http://www.digilentinc.com/Data/Products/BASYS/BASYS_C_rm.pdf

<http://www.xilinx.com/>

<http://energiasolarfotovoltaica.blogspot.com/2006/01/el-inversor.html>

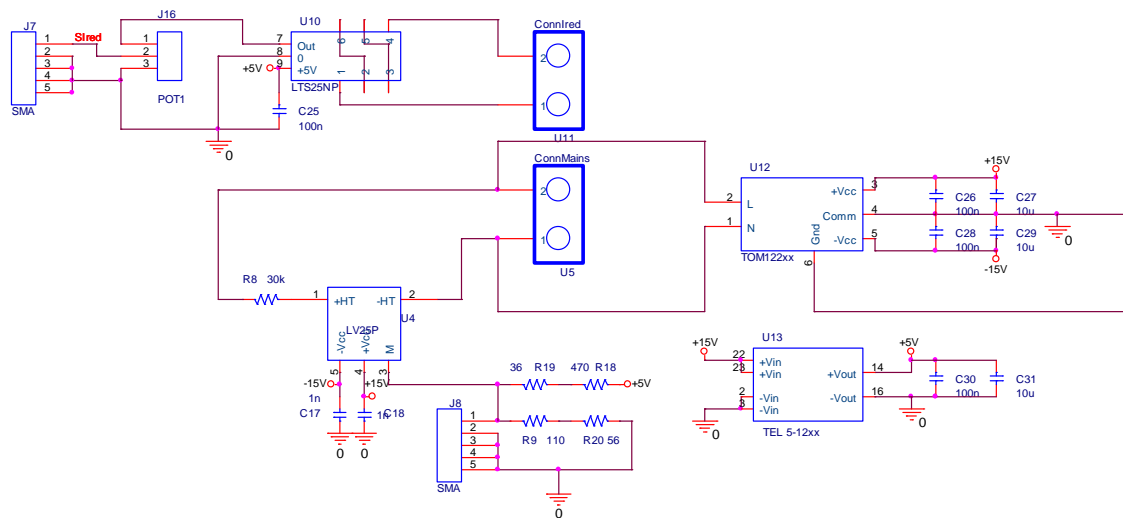
http://es.wikipedia.org/wiki/Energ%C3%ADa_solar_fotovoltaica



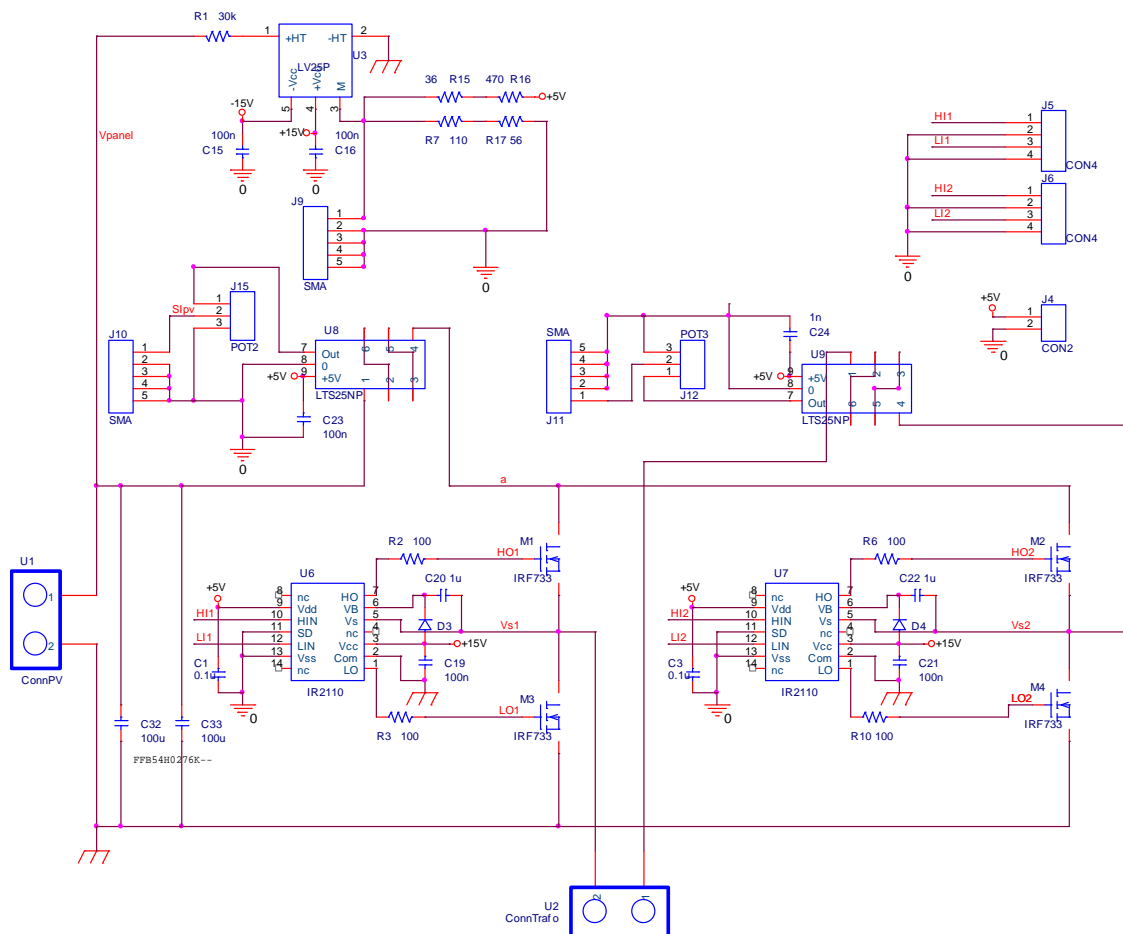
ANEXOS.

ANEXO I. ESQUEMÁTICO Y LAYOUT DE LA PCB DE LA ETAPA DE POTENCIA

ESQUEMÁTICO DE LA PCB1: HOJA 1

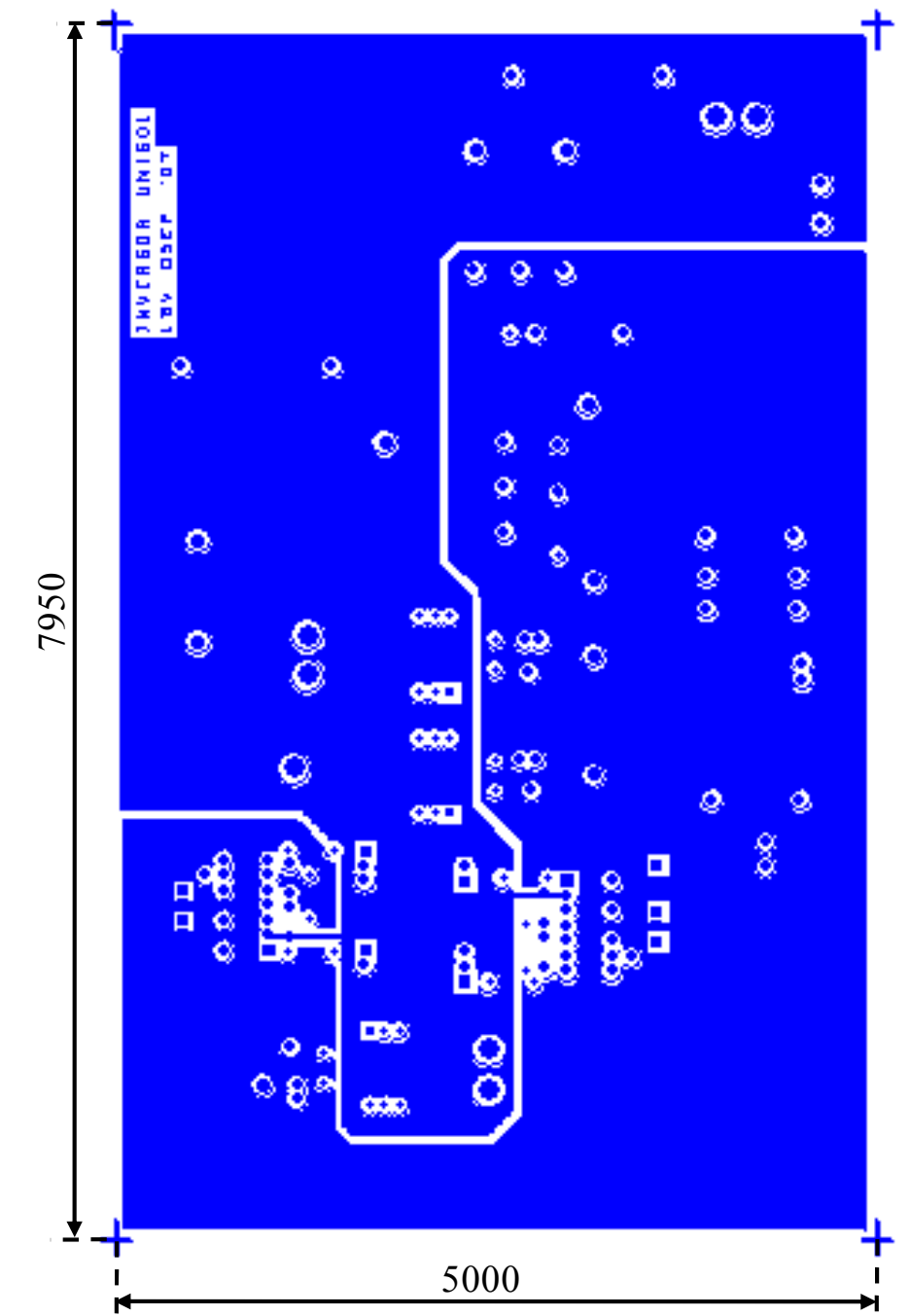


ESQUEMÁTICO DE LA PCB1: HOJA 2

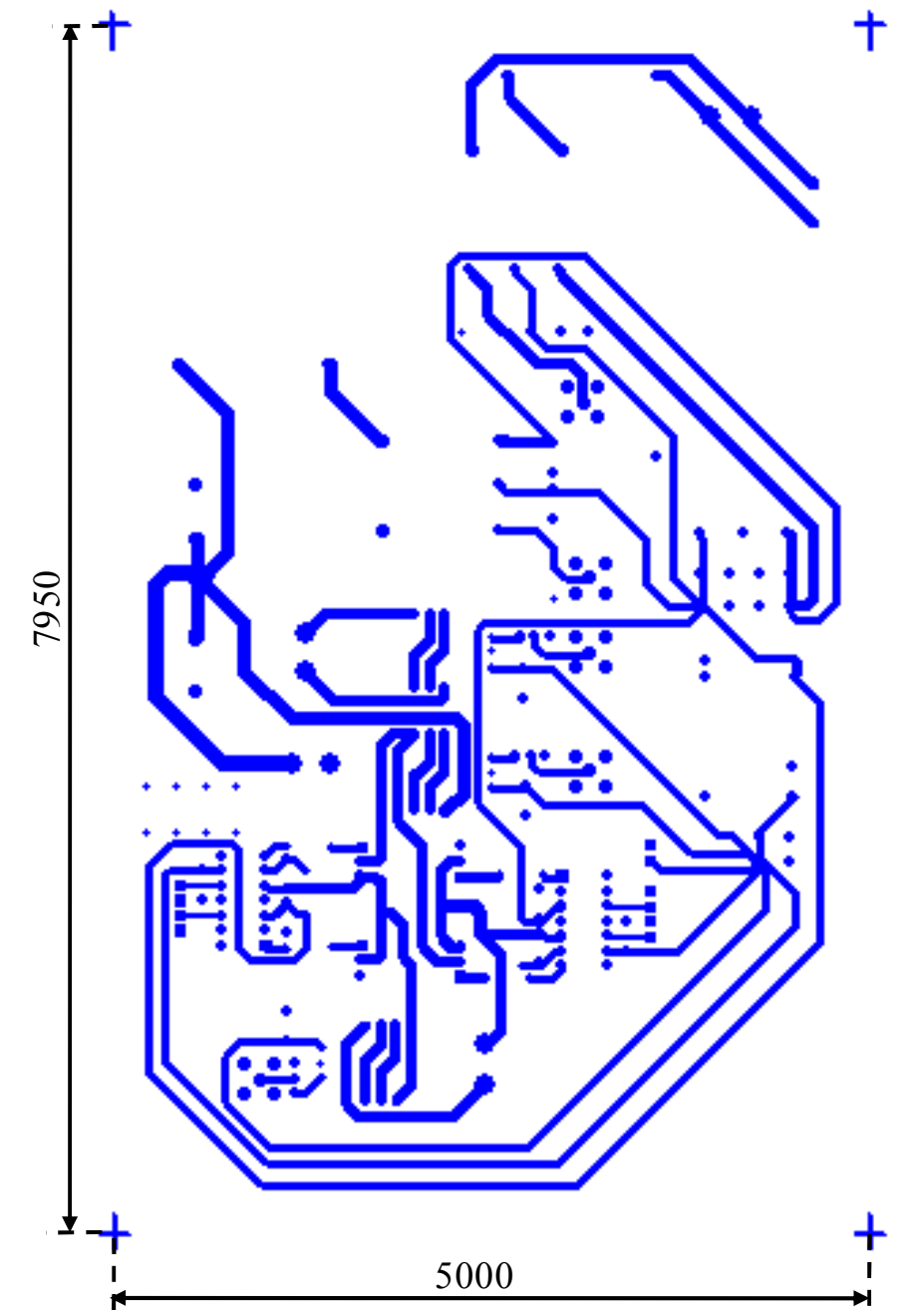


LAYOUT DE LA PCB1

- PLANO TOP

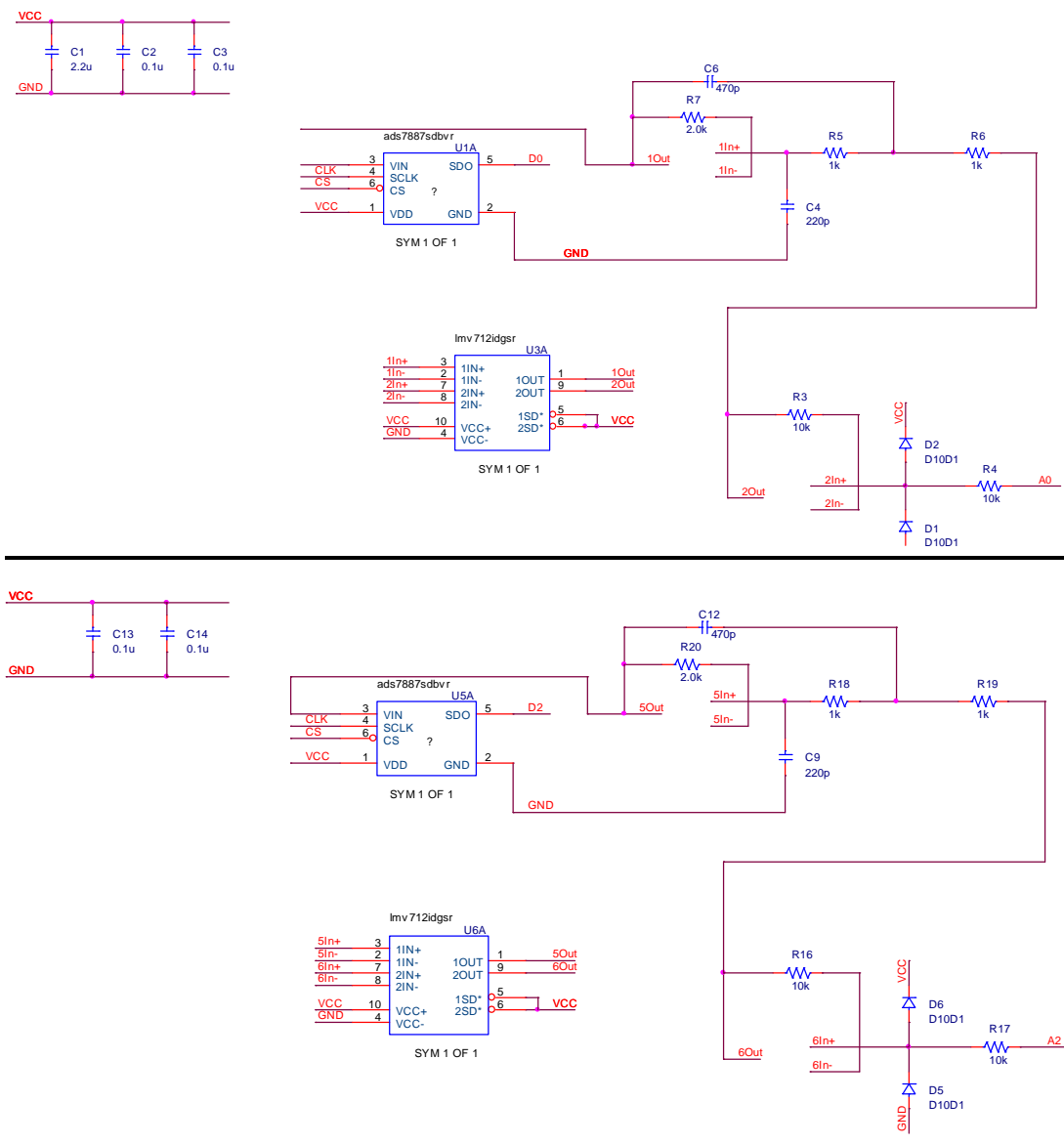


- **PLANO BOTTOM**

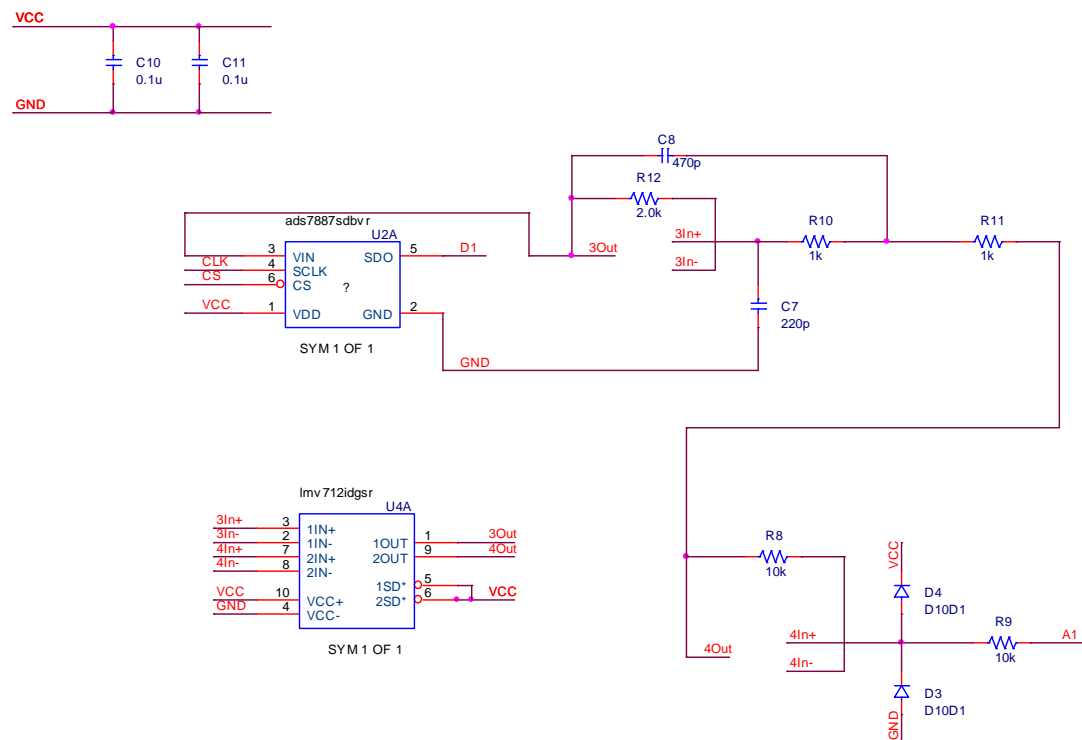
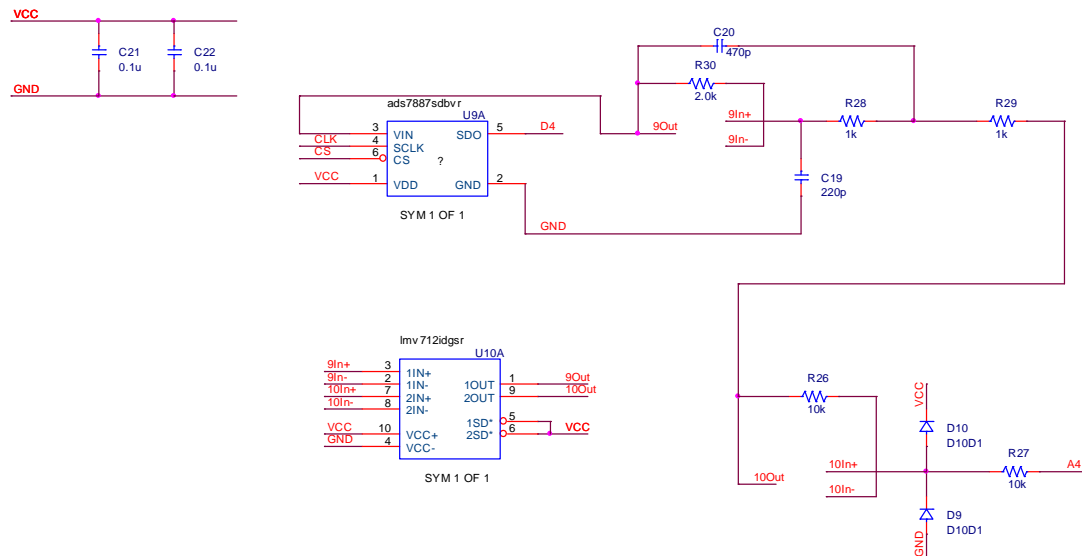


ANEXO II. ESQUEMÁTICO Y LAYOUT DE LA PCB DE LOS CONVERTIDORES A/D

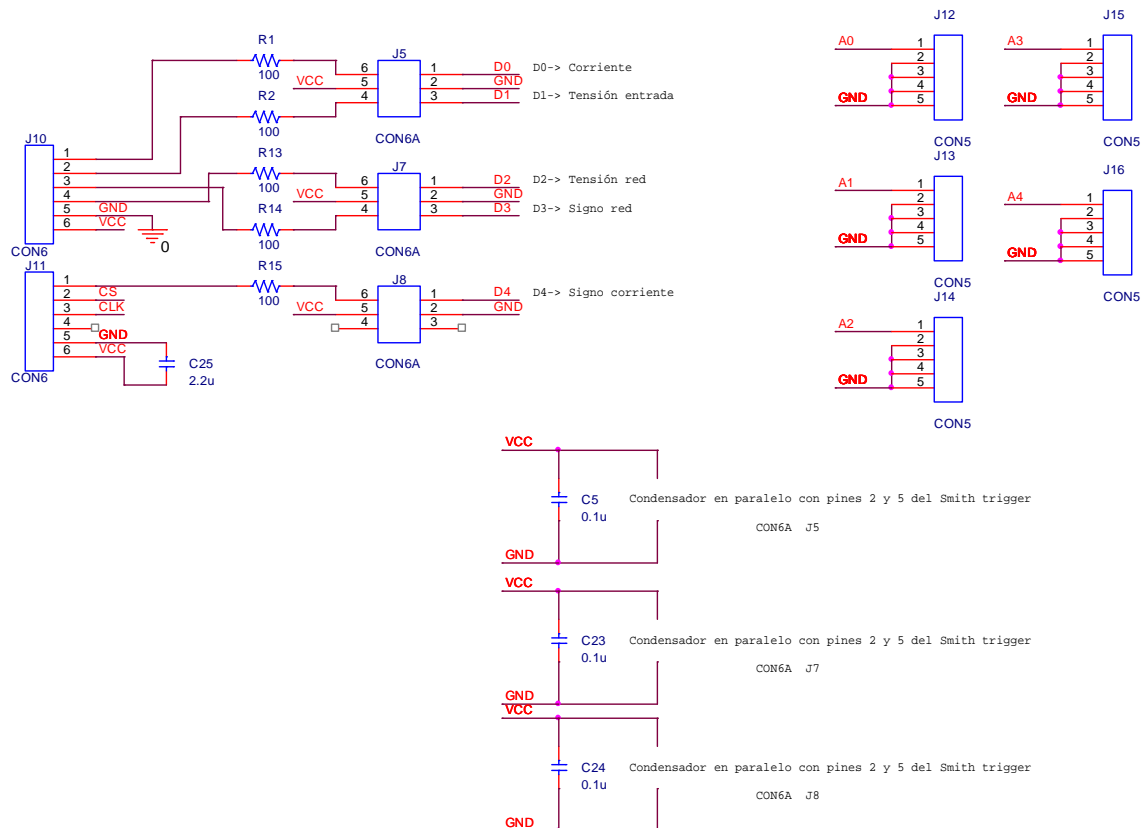
ESQUEMÁTICO DE LA PCB2: PÁGINAS 1 Y 2



ESQUEMÁTICO DE LA PCB2: PÁGINAS 2 Y 3

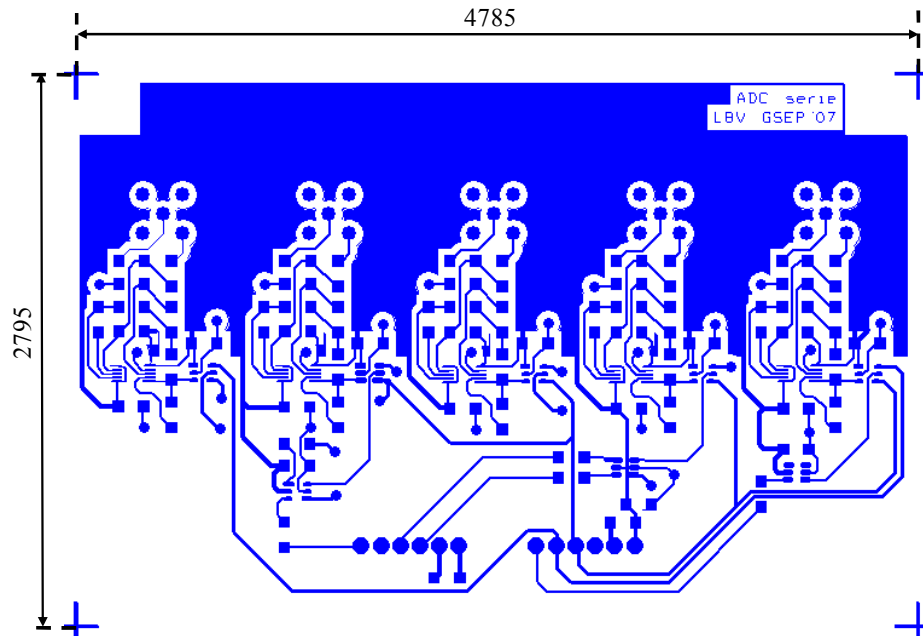


ESQUEMÁTICO DE LA PCB2: PÁGINA 4

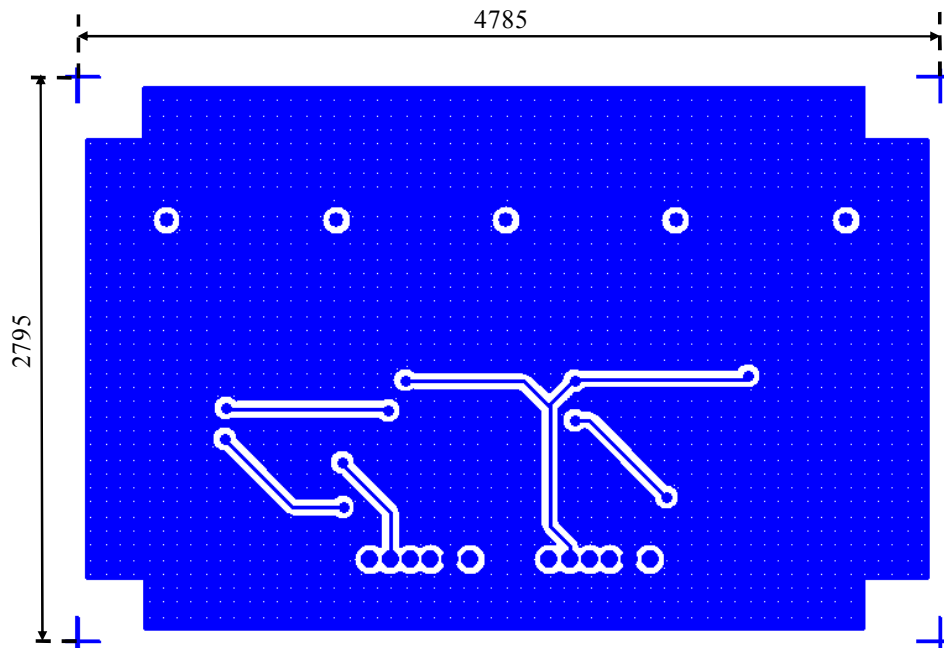


LAYOUT DE LA PCB2

- PLANO TOP



- PLANO BOTTOM



ANEXO III. CÁLCULOS CORRESPONDIENTES AL CIRCUITO DE ACONDICIONAMIENTO DE U_{RED} SENSADA

La tensión de red sensada consiste en una señal sinusoidal de 50 Hz con amplitud dependiente del valor de R_M y centrada en el origen de coordenadas. Mediante la adición del circuito de acondicionamiento del sensor mostrado en la *Figura 3.11*, se pretende convertir esta señal analógica en otra cuyos límites estén dentro del rango perceptible por el convertidor A/D.

Para calcular los valores adecuados de los componentes del circuito de acondicionamiento el primer paso es seleccionar un diodo Zener adecuado. En este caso, se ha escogido un Zener con tensión zener $V_Z = 3\text{ V}$, para el cual se ha empleado una resistencia de polarización $R_A = 82\ \Omega$.

A continuación, se puede realizar una estimación de la ganancia y el offset. Para ello, se ha partido de la condición de que la mayor parte de la corriente suministrada por el sensor, I_s , ha de circular por la resistencia de medida R_M . Esta condición se consigue imponiendo $R_{POT} \gg R_M$, con lo que se tendría que $R_{POT} + R_M \approx R_{POT}$.

De este modo, la ganancia y el offset expresados en la *Ecuación 3.3* y *Ecuación 3.4* respectivamente, quedan simplificadas como se muestra a continuación:

$$\text{Ganancia} = (1 - k_2) \cdot R_M$$

$$\text{Offset} = 3 \cdot (1 - k_2)$$

Por otra parte, se sabe que $I_{Smax} = \pm 10\text{ mA}$. Si además se tiene en cuenta la consideración de que $R_{POT} \gg R_M$ y la restricción de que $100\ \Omega \leq R_M \leq 350\ \Omega$ (según hojas de catálogo del sensor LV 25-P), se puede imponer $R_M = 220\ \Omega$, logrando de este modo los siguientes límites de tensión en la resistencia de medida, V_{RM} :

$$V_{R_M \text{ max}} \approx \pm I_{S \text{ max}} \cdot R_M \approx \pm 10\text{mA} \cdot 220\ \Omega \approx \pm 2.2\text{V}$$

Sustituyendo $V_{RM\ max}$ en la *Ecuación 3.2*, se obtienen los límites de la tensión de salida del circuito de acondicionamiento en función de la constante k_2 :

$$V_{IN_ADC\ max} = (3 + 2.2) \cdot (1 - k_2) = 5.2 \cdot (1 - k_2)$$

$$V_{IN_ADC\ min} = (3 - 2.2) \cdot (1 - k_2) = 0.8 \cdot (1 - k_2)$$

Considerando que $k_2 \leq 1$, los límites de V_{IN_ADC} tendrán como máximo unos valores de:

$$V_{IN_ADCmin} = 0.8\ V$$

$$V_{IN_ADCmax} = 5.2\ V$$

Como el valor máximo de la tensión de entrada de los ADC es $V_{IN_ADCmax} = 2.5\ V$, se puede calcular el valor de k_2 que cumpla esta restricción:

$$2.5 = 5.2 \cdot (1 - k_2) \rightarrow k_2 = 1 - \frac{2.5}{5.2} = 0.52$$

Para que k_2 sea aceptable, ha de cumplirse además que la mínima tensión de entrada del ADC sea superior a 0 V:

$$V_{IN_ADCmin} = 0.8 \cdot (1 - 0.52) = 0.384\ V \rightarrow k_2 \text{ es válido.}$$

Luego para la $k_2 = 0.52$, se tienen los siguientes valores de ganancia y offset:

$$\text{Ganancia} = (1 - 0.52) \cdot R_M = 105.6\ \frac{V}{V}$$

$$\text{Offset} = 3 \cdot (1 - k_2) = 1.44\ V$$

A la vista de los resultados obtenidos, se puede considerar como válida la aproximación de que $R_{POT} \gg R_M$, por lo cual, habiendo tomado $R_M = 220\ \Omega$, se escoge un potenciómetro de por lo menos dos órdenes de magnitud superior $\rightarrow R_{POT} = 5.1\ k\Omega$.

ANEXO IV. CÁLCULOS CORRESPONDIENTES AL CIRCUITO DE ACONDICIONAMIENTO DE I_0 SENSADA

La corriente de salida o corriente entregada a la red (i_O), responde a una corriente sinusoidal en fase con la red. En este proyecto se realizan los ensayos con corrientes máximas de ± 2 A de pico, y como para una corriente de medida $I_P = \pm 8$ A $\rightarrow V_{O_SENSOR} = 2.5 \pm 0,600$ V, se tiene que para una $I_P = \pm 2$ A, la tensión de salida del sensor será: $V_{O_SENSOR} = 2.5 \pm 0,150$ V.

De este modo, los límites de la tensión de salida del sensor para $I_P = \pm 2$ A son:

$$V_{0_SENSORmin} = 2.5 - 0.15 = 2.35 \text{ V}$$

$$V_{0_SENSORmax} = 2.5 + 0.15 = 2.65 \text{ V}$$

Por otra parte, se pueden hacer los siguientes cambios de variable en la *Ecuación 3.6* para facilitar los cálculos:

$$C_1 = \frac{R_B}{R_A + R_B} \quad C_2 = \frac{R_O}{R_1}$$

Con este cambio de variable, la *Ecuación 3.6* puede expresarse de forma simplificada:

$$V_O = V_{O_{sensor}} \cdot C_1 \cdot (1 + C_2) - V_R \cdot C_2$$

Donde se sabe que $C_1 \leq 1$ y que los límites de V_{O_SENSOR} son 2.35 V y 2.65 V.

Con estas condiciones, se pretende encontrar unos valores adecuados de C_1 y C_2 con los que los límites de la tensión de salida V_O estén entre 0 y 2.5 V.

Para simplificar el problema, se impone la restricción de que $C_1 = 1$, lo que supondría que o bien $R_A = 0$ o $R_B = \infty$. Como R_A forma parte del filtro antialiasing inicial, se puede suprimir R_B del diseño, quedando por calcular únicamente el valor de R_1 .



Con las restricciones impuestas es posible encontrar un valor de C_2 que imponga unos límites de V_O entre 0 y 2.5 V.

Suponiendo $V_{Omin} = 0.1 \text{ V}$, se tiene:

$$0.1 \text{ V} = 2.35 \cdot 1 \cdot (1 + C_2) - 3.3 \cdot C_2; \quad C_2 = 2.34$$

Para este valor de C_2 se puede comprobar que V_{Omax} cumple con la restricción $V_{Omax} \leq 2.5 \text{ V}$:

$$V_{Omax} = 2.65 \cdot 1 \cdot (1 + 2.34) - 3.3 \cdot 2.34 = 1.127 \text{ V} < 2.5 \text{ V}$$

De este modo, se considera como válido el valor hallado de C_2 , obteniendo el siguiente valor de la resistencia R_1 :

$$R_1 = \frac{R_O}{C_2} = \frac{10000}{2.34} = 4.27 \text{ k}\Omega$$



ANEXO V. SCRIPT REALIZADO CON MATLAB PARA LA GENERACIÓN DE TABLAS DE DISPARO

```
%PWM generada para un puente completo de transistores
%conmutando una rama a alta frecuencia y la otra a baja
%frecuencia en cada semiciclo de red
%En cada semiciclo se alternan las ramas

%Autor: Pablo Zumel
%Fecha: Julio 2007
%Revision: Octubre 2007

%Calcula los angulos suponiendo una resolución de 1e6 puntos
%por periodo de red, lo que corresponde a
%50Hz de frecuencia de red
%50MHz de frecuencia de reloj de la FPGA

%Frecuencia de conmutación de la PWM e índice de modulación
fs=10000;
ma=0.8

%Numero periodos de la PWM incluidos en un ciclo de red
k=fs/50;

%Coordenada x para el cálculo de la PWM (ángulo)
t=linspace(0,2*pi,1e06);

%Función moduladora (seno)
seno=ma*sin(t);

%Función Portadora (diente de sierra)
%generada para la conmutación a baja frecuencia
%de una de las ramas

%diente de sierra del semiciclo positivo
s1=0.5*sawtooth(k*t,0.5)+0.5;
%diente de sierra del semiciclo negativo
s2=-0.5*sawtooth(k*t,0.5)-0.5;
%Diente de sierra total
s=[s1(1,1:500000), s2(1,500001:1000000)];

%Generación de los pulsos
%Vector que vale 1 cuando se cumple la condición s<=seno, es decir, la
%portadora es menor que la moduladora
pulsosp=(s<=seno);
pulsosn=(s>=seno);

pulsos=[pulsosp(1:500000) pulsosn(500001:1000000)];

%Vector de pulsos desplazado un punto (el vector total tiene 1e6
puntos)
pulsosb=circshift(pulsos,[1,-1]);
```



```
%Flancos es un vector que contiene un 1 en los puntos en los que
"pulsos"
%cambia, es decir, se detectan los flancos de "pulsos"
flancos=xor(pulsos(1,1:250000),pulsosb(1,1:250000));
%Genera un vector correspondiente a los puntos en los que hay un
flanco, es
%decir, genera el vector con los puntos en los que "pulsos" ha
cambiado
%Lo genera por cuartos
ang_aux=find(flancos)'; %primer cuarto
anga=ang_aux(2:size(ang_aux,1)); %Se elimina el primer elemento,
porque siempre sale 1 y no tiene sentido
angb=flipud(500000-anga); %segundo cuarto

%Resultado final
%Vector que contiene los instantes de cambio
%suponiendo que el periodo esta dividido en 1e6 puntos.
ang=[anga;angb;anga+500000;angb+500000];

%Genera el vector de los valores que tiene que tomar el pulso de
control al
%pasar por el instante de cambio correspondiente
%Esto es para evitar que se pierda en la generación de los pulsos y
genere
%los negados
value=zeros(size(ang));
for i=1:size(ang,1)
    if (i/2)==floor(i/2)
        value(i,1)=0;
    else value(i,1)=1;
    end
end

%Cálculo de los armónicos para comprobar el resultado

%Cálculo de la tensión de salida del inversor
Vab=[pulsos(1:5*1e5) -1*pulsos(500001:1000000)];
F=fft([Vab]);
Pyy=abs(F)/1e6;%F.*conj(F)/1e6;
f=(1/20e-9)*(0:500000)/1e6;

%Visualización de los resultados
figure
subplot(211)
plot(seno);
hold on;
plot(s,'r');
subplot(212)
plot(Vab)
hold on
stem(ang,value)

figure
stem(f(1:2000),Pyy(1:2000))
```

ANEXO VI. DESCRIPCIÓN VHDL PARA LA GENERACIÓN DE UNA PWM UNIPOLAR

CÓDIGO SINTETIZABLE Y SIMULABLE

- *PWM_UNIPOLAR.VHD*

```
-----
--          Universidad Carlos III de Madrid          --
--          Departamento de Tecnología Electrónica    --
--          Grupo de Sistemas Electrónicos de Potencia --
-----
--          GENERADOR DE ONDA UNIPOLAR PARA INVERSORES      --
-----
-- Autor: Pablo Zumel                                     --
-- Fecha de creacion: 03/07/2007                          --
-- Fecha de la ultima revision: 27/10/2007                --
-----

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity PWM_unipolar is
Port ( Clk      : in std_logic;
      ResN      : in std_logic;
      Tr4       : out std_logic; --CyclePosT14 : out std_logic;
      Tr3       : out std_logic; --CycleNegT23 : out std_logic;
      Tr2       : out std_logic; --PWM_23      : out std_logic;
      Tr1       : out std_logic; --PWM_14      : out std_logic;
      sel_freq  : in std_logic_vector (3 downto 0)
    );
end PWM_unipolar;

architecture Behavioral of PWM_unipolar is

component param_table      --configuracion tablas

port ( Clk      : in std_logic;
      ADDR      : in integer range 0 to 14;
      DOUT      : out integer range 0 to 1023
    );
```



```
end component;

component table_1

  port ( Clk      : in std_logic;
        ADDR     : in integer range 0 to 1023;
        DOUT     : out integer range 0 to 999999;
        LOUT     : out integer range 0 to 1
      );
end component;

component table_2

  port ( Clk      : in std_logic;
        ADDR     : in integer range 0 to 1023;
        DOUT     : out integer range 0 to 999999;
        LOUT     : out integer range 0 to 1
      );
end component;

component table_3

  port ( Clk      : in std_logic;
        ADDR     : in integer range 0 to 2047;
        DOUT     : out integer range 0 to 999999;
        LOUT     : out integer range 0 to 1
      );
end component;

constant numero_frecs : integer:=15;
type int_vector      is array (numero_frecs-1 downto 0) of integer range 0 to 999999;
type bit_int_vector is array (numero_frecs-1 downto 0) of integer range 0 to 1;

constant tiempo_muerto : integer :=50;

signal sel_frec_int  : integer range 0 to 15;
signal time          : integer range 0 to 999999;
signal en            : std_logic;
signal pointer_max   : integer range 0 to 8191;
signal pointer       : integer range 0 to 8191:=0; -- este es la longitud máxima de la
signal tableIN_Aux   : integer;                  -- tabla, que corresponde con una
signal angle         : integer range 0 to 999999; -- PWM de hasta 200 kHz
signal angle_vector  : int_vector;
signal value         : integer ;
signal value_in      : bit_int_vector :=(others=>0);
signal ADDR_vector   : int_vector :=(others=>0);
signal pwm_out       : std_logic_vector (4 downto 1);
```



```
signal estado      : std_logic;
signal angle_anterior : integer range 0 to 999999;
signal pulso_tm     : std_logic;

begin

-----
--  Instanciacion de las tablas      --
-----
-- ADDR_vector contiene la dirección que se quiere leer
-- de cada una de las tablas

-- value_in_i contienen el valor del pulso correspondiente a cada ángulo

-- Las tablas se leen permanentemente, es decir, el valor de angle_vector
-- value_in son señales que en todo momento son el valor de la tabla para
-- la dirección correspondiente
-- Si se quiere leer otro dato basta con cambiar la dirección de las tablas
-- La "lectura" de las tablas se hace mediante la simple asignación de puertos

DOUT=>value_in

tableParam: param_table

    port map ( Clk    => Clk,
               ADDR   => sel_freq_int,
               DOUT   => pointer_max
             );
table1: table_1

    port map ( Clk    => Clk,
               ADDR   => ADDR_vector(0),
               DOUT   => angle_vector(0),
               LOUT   => value_in(0)
             );

table2: table_2

    port map ( Clk    => Clk,
               ADDR   => ADDR_vector(1),
               DOUT   => angle_vector(1),
               LOUT   => value_in(1)
             );

table3: table_3

    port map ( Clk    => Clk,
```



```
ADDR  => ADDR_vector(2),
DOUT  => angle_vector(2),
LOUT  => value_in(2)
);
```

```
-----
--    Contador PRINCIPAL                                --
-----
```

```
Contador_principal: process (ResN,Clk)
begin
```

```
    if ResN='0' then
        time<=0;
    elsif Clk='1' and Clk'event then
        if time < 999999 then
            time <= time+1;
        else time <= 0;
        end if;
    end if;
end process Contador_principal;
```

-- Señal de sincronismo que se genera en el primer ciclo de reloj del ciclo de red

```
enable_principal: process (ResN,Clk)
begin
```

```
    if ResN='0' then
        en<='0';
    elsif Clk='1' and Clk'event then
        if time=999999 then
            en<='1';
        else en<='0';
        end if;
    end if;
end process enable_principal;
```

```
-----
--    Generación de la selección de tablas                --
--    Para el caso normal(un único patrón) y para la    --
--    PWM con dos frecuencias alternadas                  --
-----
```

```
Cambiotable: process(Clk,ResN)
begin
    if ResN='0' then
```




```
sel_frec_int<=0;
elsif Clk='1' and Clk'event and en='1' then
  if sel_frec = "0011" then
    estado<= not estado;
    if estado='0' then
      sel_frec_int<=0;
    elsif estado='1' then
      sel_frec_int<=3;
    end if;
  else sel_frec_int<=conv_integer(sel_frec);
  end if;
end if;
end process Cambiotable;

-----
--  Lectura table  --
-----

-- Se asigna a la señal angle correspondiente a
-- la frecuencia seleccionada del valor del vector
-- angle_vector que contiene los valores leídos
-- de todos las frecuencias

lectura_tabla: process (Clk,ResN)
begin
  if ResN='0' then
    angle<=0;
    value<=0;
  elsif Clk='1' and Clk'event then
    angle<=angle_vector(sel_frec_int);
    case sel_frec_int is
      when 0 => value<=value_in(0);
      when 1 => value<=value_in(1);
      when 2 => value<=value_in(2);
      when 3 => value<=value_in(3);
      when others => value<=value_in(0);
    end case;
  end if;
end process lectura_tabla;

-- Cada vez
avance_pointer: process (Clk,ResN)
begin
  if ResN='0' then
    pointer<=0;
  elsif Clk='1' and Clk'event then
    if angle=time then
```



```
-- nuevo
angle_anterior<=angle;
-- nuevo
if pointer=pointer_max then
    pointer<=0;
else pointer<=pointer+1;
end if;
end if;

-- nuevo
if time>(angle-tiempo_muerto) or time < (angle_anterior+tiempo_muerto) then
    pulso_tm<='0';
else pulso_tm<='1';
end if;
-- nuevo
end if;
end process avance_pointer;
```

-- Eleccion de TABLA --

```
eleccion_tabla: process(Clk,ResN)
begin
    if ResN='0' then
        for i in numero_frecs-1 downto 0 loop
            ADDR_vector(i)<=0;
        end loop;
    elsif Clk='1' and Clk'event then
        for i in numero_frecs-1 downto 0 loop
            if i= sel_frec_int then
                ADDR_vector(i)<=pointer;
            else ADDR_vector(i)<=0;
            end if;
        end loop;
    end if;
end process eleccion_tabla;
```

-- Generación de disparos --

```
PWM: process (Clk,ResN)
begin
    if ResN='0' then
        pwm_out<=(others=>'0');
    elsif Clk='1' and Clk'event then
```



```
if time< 500000 then
    pwm_out(2)<='0';
    pwm_out(4)<='1';
    if value=1 then
        pwm_out(1)<='1';
        pwm_out(3)<='0';
    else
        pwm_out(1)<='0';
        pwm_out(3)<='1';
    end if;
elseif time> 500000 or time= 500000 then
    pwm_out(1)<='0';
    pwm_out(3)<='1';
    if value=1 then
        pwm_out(2)<='1';
        pwm_out(4)<='0';
    else
        pwm_out(2)<='0';
        pwm_out(4)<='1';
    end if;
end if;
end if;
end process PWM;

Tr4    <=pwm_out(4);
Tr3    <=pwm_out(3);
--nuevo
Tr2    <=pwm_out(2) and pulso_tm;
Tr1    <=pwm_out(1) and pulso_tm;
--nuevo

end Behavioral;
```

- *PARAM_TABLE.VHD*

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

-- Uncomment the following lines to use the declarations that are
-- provided for instantiating Xilinx primitive components.
--library UNISIM;
--use UNISIM.VComponents.all;

entity param_table is
```



```
Port ( Clk : in std_logic;
      ADDR : in integer range 0 to 15;
      DOUT : out integer range 0 to 1023
    );

end param_table;

architecture Behavioral of param_table is
  type matrice is array (0 to 15) of integer range 0 to 1023;
  constant parametro:matrice :=(

    -- ***** Ma=0.8 ***** --

1195 ,      --      30000
795 ,      --      20000
395 ,      --      10000
63 ,       --      1700
59 ,       --      8000
59 ,       --      7142,857143
59 ,       --      6250
59 ,       --      5000
59 ,       --      4545,454545
59 ,       --      4000
59 ,       --      3030,30303
59 ,       --      2500
59 ,       --      2000
59 ,       --      1600
59 ,       --      1503,759398
59
    -- ***** --
  );

begin
  Process(Clk)
  begin
    if Clk'event and Clk='1' then
      DOUT <= parametro(ADDR);
    end if;
  end process;
end Behavioral;
```

- *TABLE1.VHD*



--	Universidad Carlos III de Madrid	--
--	Departamento de Tecnología Electrónica	--
--	Grupo de Sistemas Electrónicos de Potencia	--

--	GENERADOR DE ONDA UNIPOLAR PARA INVERSORES	--

--	Autor: Pablo Zumel	--
--	Fecha de creacion: 03/07/2007	--
--	Fecha de la ultima revision: 27/10/2007	--

library IEEE;		
use IEEE.STD_LOGIC_1164.ALL;		
use IEEE.STD_LOGIC_ARITH.ALL;		
use IEEE.STD_LOGIC_UNSIGNED.ALL;		
entity table_1 is		
Port (Clk : in std_logic;		
ADDR : in integer range 0 to 1023;		
DOUT : out integer range 0 to 999999;		
LOUT : out integer range 0 to 1		
);		
end table_1;		
architecture Behavioral of table_1 is		
type matrice_angulo1 is array (0 to 395) of integer range 0 to 999999;		
type matrice_bit1 is array (0 to 395) of integer range 0 to 1;		
constant angulos1: matrice_angulo1 :=(
-- ***** Ma=0.8 y Fc=10 KHz***** --		
--		
-- Espacio reservado para los ángulos de disparo		
-- obtenidos mediante el script de Matlab. Esta tabla tiene		
-- reservadas 396 posiciones, resultantes de una Fconmutacion= 10 KHz y Ma= 0.8		
--		
-- ***** --		
);		
constant nivel1: matrice_bit1 :=(
-- ***** Ma=0.8 y Fc=10 KHz ***** --		
--		
-- Espacio reservado para los valores digitales de disparo		
-- obtenidos mediante el script de Matlab. Esta tabla tiene		



```
-- reservadas 396 posiciones, resultantes de una Fconmutacion= 10 KHz y Ma= 0.8
--
-- *****
);

begin
  Process(Clk)
  begin
    if Clk'event and Clk='1' then
      DOUT <= angulos1(ADDR);
      LOUT <= nivel1(ADDR);
    end if;
  end process;
end Behavioral;
```

- *TABLE2.VHD*

```
-----
--          Universidad Carlos III de Madrid          --
--          Departamento de Tecnología Electrónica    --
--          Grupo de Sistemas Electrónicos de Potencia --
-----
--          GENERADOR DE ONDA UNIPOLAR PARA INVERSORES  --
-----
-- Autor: Pablo Zumel                                --
-- Fecha de creacion: 03/07/2007                      --
-- Fecha de la ultima revision: 27/10/2007            --
-----

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity table_2 is
  Port ( Clk : in std_logic;
        ADDR : in integer range 0 to 1023;
        DOUT : out integer range 0 to 999999;
        LOUT : out integer range 0 to 1
        );
end table_2;

architecture Behavioral of table_2 is
```

```

type matrice_angulo2 is array (0 to 795) of integer range 0 to 999999;
type matrice_bit2 is array (0 to 795) of integer range 0 to 1;

constant angulos2: matrice_angulo2 :=(

-- ***** Ma=0.8 y Fc= 20 KHz ***** --
--
-- Espacio reservado para los ángulos de disparo
-- obtenidos mediante el script de Matlab. Esta tabla tiene
-- reservadas 796 posiciones, resultantes de una Fconmutacion= 20 KHz y Ma= 0.8
--
-- ***** --
);

constant nivel2: matrice_bit2 :=(

-- ***** Ma=0.8 y Fc= 20 KHz***** --
--
-- Espacio reservado para los valores digitales de disparo
-- obtenidos mediante el script de Matlab. Esta tabla tiene
-- reservadas 796 posiciones, resultantes de una Fconmutacion= 20 KHz y Ma= 0.8
--
-- ***** --
);

begin
  Process(Clk)
  begin
    if Clk'event and Clk='1' then
      DOUT <= angulos2(ADDR);
      LOUT <= nivel2(ADDR);
    end if;
  end process;
end Behavioral;

```

- *TABLE3.VHD*

```

-----
-- Universidad Carlos III de Madrid --
-- Departamento de Tecnología Electrónica --
-- Grupo de Sistemas Electrónicos de Potencia --
-----
-- GENERADOR DE ONDA UNIPOLAR PARA INVERSORES --
-----
-- Autor: Pablo Zumel --

```



```
-- Fecha de creacion: 03/07/2007 --
-- Fecha de la ultima revision: 27/10/2007 --
-----

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity table_3 is

    Port ( Clk : in std_logic;
          ADDR : in integer range 0 to 2047;
          DOUT : out integer range 0 to 999999;
          LOUT : out integer range 0 to 1
        );

end table_3;

architecture Behavioral of table_3 is
    type matrice_angulo3 is array (0 to 1195) of integer range 0 to 999999;
    type matrice_bit3 is array (0 to 1195) of integer range 0 to 1;

    constant angulos3: matrice_angulo3 :=(

-- ***** Ma=0.8 y Fc= 30 kHz***** --
--
-- Espacio reservado para los ángulos de disparo
-- obtenidos mediante el script de Matlab. Esta tabla tiene
-- reservadas 1196 posiciones, resultantes de una Fconmutacion= 30 KHz y Ma= 0.8
--
-- ***** --
    );

    constant nivel3: matrice_bit3 :=(

-- ***** Ma=0.8 y Fc= 30 kHz***** --
--
-- Espacio reservado para los valores digitales de disparo
-- obtenidos mediante el script de Matlab. Esta tabla tiene
-- reservadas 1196 posiciones, resultantes de una Fconmutacion= 30 KHz y Ma= 0.8
--
-- ***** --
    );

begin
    Process(Clk)
```




```
begin
  if Clk'event and Clk='1' then
    DOUT <= angulos3(ADDR);
    LOUT <= nivel3(ADDR);
  end if;
end process;

end Behavioral;
```



CÓDIGO SIMULABLE

- BANCO DE PRUEBAS: *PWM_UNIPOLAR_TB.VHD*

```
-----
-- TestBench --
-- Proyecto: PWM unipolar --
-- Fichero: PWM_unipolar_tb.vhd --
-- --
-- Descripcion: testbench para el modulador PWM --
-- --
-----
-- Autor: Pablo Zumel, Cristina Fernandez --
-- Fecha: 23/01/07 --
-- Fecha de ultima revision: 28/10/07 --
-----
-- UC3M-GSEP --
-----

library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity PWM_unipolar_tb is
end PWM_unipolar_tb;

architecture behavioral of PWM_unipolar_tb is

    component PWM_unipolar
    port( Clk      : in std_logic;
          ResN     : in std_logic;
          Tr4      : out std_logic;
          Tr3      : out std_logic;
          Tr2      : out std_logic;
          Tr1      : out std_logic;
          sel_freq  : in std_logic_vector (3 downto 0)
    );
    end component;

    component Invertermodel
    port(Ue      : in real :=0.0;
          Ured    : in real :=0.0;
          OnOff1SM : in std_logic; -- On = '1', off = '0'
          OnOff2SM : in std_logic; -- On = '1', off = '0'
          OnOff3SM : in std_logic; -- On = '1', off = '0'
```

```

        OnOff4SM : in std_logic; -- On = '1', off = '0'
        ResN      : in std_logic;
        Ie        : out real :=0.0;
        Io        : out real :=0.0;
        Uab       : out real :=0.0
    );
end component;

signal Clk_tb      : std_logic;
signal sel_freq_tb : std_logic_vector (3 downto 0);
signal Tr4_tb      : std_logic;
signal Tr3_tb      : std_logic;
signal ResN_tb     : std_logic;
signal Tr2_tb      : std_logic;
signal Tr1_tb      : std_logic;

signal Ue_tb       : real:=100.0;
signal Ured_tb     : real :=0.0;
signal Ie_tb       : real:=0.0;
signal Io_tb       : real:=0.0;
signal Uinv_tb     : real;

begin

    ThePWM_unipolar: PWM_unipolar port map(
        Clk => Clk_tb,
        ResN => ResN_tb,
        sel_freq => sel_freq_tb,
        Tr4 => Tr4_tb,
        Tr3 => Tr3_tb,
        Tr2 => Tr2_tb,
        Tr1 => Tr1_tb
    );

    TheInverter_model: Invertermodel port map(
        Ue => Ue_tb,
        Ured => Ured_tb,
        OnOff1SM => Tr1_tb,
        OnOff2SM => Tr2_tb,
        OnOff3SM => Tr3_tb,
        OnOff4SM => Tr4_tb,
        ResN => ResN_tb,
        Ie => Ie_tb,
        Io => Io_tb,
        Uab => Uinv_tb
    );

```



```
sel_freq_generator: process
begin
sel_freq_tb<= "0000";
wait for 20 ms;

sel_freq_tb<= "0001";
wait for 20 ms;

sel_freq_tb<= "0010";
wait for 20 ms;

sel_freq_tb<= "0011";
wait for 80 ms;

end process;

Clock_generator : PROCESS
begin
wait for 10 ns;
Clk_tb <= '1';
wait for 10 ns;
Clk_tb <= '0';
end process Clock_generator;

ResN_generator : process
begin
ResN_tb <= '0';
wait for 500 ns;
ResN_tb <= '1';
wait;
end process;

end;
```

- MODELO DE INVERSOR: *INVERTERMODEL.VHD*

```
-- Fichero: InverterModel.vhd
--
-- Descripción: Modelo simple de un inversor conectado a red.
-- Se considera una bobina en serie con la fuente de tensión
-- que representa la tensión de la línea
-----
-- Autor: Cristina Fernandez y Pablo Zumel
-- Fecha: 04/02/05
-----
-- UC3M-GSEP
-----
```



```
library WORK, IEEE;
use IEEE.std_logic_1164.all;

entity InverterModel is
port( Ue   : in real :=100.0;
      Ured  : in real :=0.0;
      OnOff1SM : in std_logic; -- On = '1', off = '0'
      OnOff2SM : in std_logic; -- On = '1', off = '0'
      OnOff3SM : in std_logic; -- On = '1', off = '0'
      OnOff4SM : in std_logic; -- On = '1', off = '0'
      ResN     : in std_logic;
      Ie       : out real :=0.0;
      Io       : out real :=0.0;
      Uab      : out real :=0.0
    );
end InverterModel;

-----

-- Significado de los puertos:
-- Ue: tension continua de entrada
-- Ured: tension senoidal de la red
-- OnOff1SM: interruptor 1 ON a '1' y OFF a '0'
-- OnOff2SM: interruptor 1 ON a '1' y OFF a '0'
-- OnOff3SM: interruptor 1 ON a '1' y OFF a '0'
-- OnOff4SM: interruptor 1 ON a '1' y OFF a '0'
-- ResN: reset por nivel bajo, inicializa Il1 y Us
-- Ie: intensidad de entrada
-- Is: intensidad de salida
-- Uab: tension de salida del puente inversor

-----

architecture Comportamental of InverterModel is

    constant cycle : time:=100 ns;
    constant dt    : real:=100.0e-9;
    constant L     : real:=800.0e-6;
    constant R     : real:=1.0;

    signal IeAux : real :=0.0;-- Auxiliar de Ie para poder leerse
    signal IoAux : real :=0.0;-- Auxiliar de Is para poder leerse
    signal UabAux : real :=0.0;-- Auxiliar de Uab para poder leerse

begin

    Ie <= IeAux;
```

```
Io <= IoAux;
Uab <= UabAux;

CALC: process
-- Cada periodo de integracion se dan nuevos valores a Is e Ie
begin
  if ResN = '0' then -- inicializacion Is
    UabAux <= 0.0;
    elsif OnOff1SM = '1' and OnOff2SM='0' and OnOff3SM = '0' and OnOff4SM='1'
  then -- se aplica la tension positiva a la carga
    UabAux <= Ue;
    IeAux <= IoAux;
    elsif OnOff1SM = '0' and OnOff2SM='1' and OnOff3SM = '1' and OnOff4SM='0'
  then -- se aplica la tension negativa a la carga
    UabAux <= -Ue;
    IeAux <= -IoAux;
    elsif OnOff1SM = '1' and OnOff2SM='1' and OnOff3SM = '0' and OnOff4SM='0'
  then -- se cortocircuita la carga
    UabAux <= 0.0;
    IeAux <= 0.0;
    elsif OnOff1SM = '0' and OnOff2SM='0' and OnOff3SM = '1' and OnOff4SM='1'
  then -- se cortocircuita la carga
    UabAux <= 0.0;
    IeAux <= 0.0;
    elsif OnOff1SM = '0' and OnOff2SM='0' and OnOff3SM = '0' and OnOff4SM='0'
  then
    if IoAux < 0.0 then -- la corriente circula por los diodos parásitos de 1 y 4
      UabAux <= -Ue;
      IeAux <= -IoAux;
    else -- la corriente circula por los diodos parásitos de 2 y 3
      UabAux <= Ue;
      IeAux <= IoAux;
    end if;
    -- Analisis de los tiempos muertos
    elsif OnOff1SM = '1' and OnOff2SM='0' and OnOff3SM = '0' and OnOff4SM='0'
  then
    if IoAux < 0.0 then -- la corriente circula por el diodo parasito 4
      UabAux <= Ue;
      IeAux <= IoAux;
    else -- la corriente circula por el diodo parasito 2
      UabAux <= 0.0;
      IeAux <= 0.0;
    end if;
    elsif OnOff1SM = '0' and OnOff2SM='1' and OnOff3SM = '0' and OnOff4SM='0'
  then
    if IoAux < 0.0 then -- la corriente circula por el diodo parasito 1
```



```
UabAux <= 0.0;
IeAux <= 0.0;
else -- la corriente circula por el diodo parasito 3
    UabAux <= -Ue;
    IeAux <= -IoAux;
end if;
elsif OnOff1SM = '0' and OnOff2SM='0' and OnOff3SM = '1' and OnOff4SM='0'
then
    if IoAux < 0.0 then -- la corriente circula por el diodo parasito 4
        UabAux <= 0.0;
        IeAux <= 0.0;
    else -- la corriente circula por el diodo parasito 2
        UabAux <= -Ue;
        IeAux <= -IoAux;
    end if;
    elsif OnOff1SM = '0' and OnOff2SM='0' and OnOff3SM = '0' and OnOff4SM='1'
then
    if IoAux < 0.0 then -- la corriente circula por el diodo parasito 1
        UabAux <= Ue;
        IeAux <= IoAux;
    else -- la corriente circula por el diodo parasito 3
        UabAux <= 0.0;
        IeAux <= 0.0;
    end if;
    else -- este caso no deberia darse
        UabAux <= 0.0;
        IeAux <= 0.0;
    end if;
    IoAux <= IoAux + ((UabAux-Ured-IoAux*R)/L)*dt;
    wait for cycle;
end process CALC;

end Comportamental;
```




ANEXO VII. DESCRIPCIÓN VHDL PARA EL CONTROL DE LOS CONVERTIDORES A/D

CÓDIGO SINTETIZABLE Y SIMULABLE

- *ADC_MODULO.VHD*

```
-----
-- Company: UC3M
-- Engineer: Leandro E. Boyano Vannucchi
-- Create Date: 15:35:28 10/04/2007
-- Design Name:
-- Module Name: ADC_modulo - Behavioral
-- Project Name:
-- Target Devices:
-- Tool versions:
-- Description:
-- Dependencies:
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
-----

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
use IEEE.NUMERIC_STD.ALL;

---- Uncomment the following library declaration if instantiating
---- any Xilinx primitives in this code.
---- library UNISIM;
---- use UNISIM.VComponents.all;

entity AD_serial_converter is

    port(
        Conv_ST      : in std_logic;
--      Switch      : in std_logic_vector (0 to 4); -- Selector de canal para probar cada
        clk          : in std_logic;                -- canal independientemente del resto
        Reset        : in std_logic;
        DATA1       : in std_logic;
        DATA2       : in std_logic;
        DATA3       : in std_logic;
        DATA4       : in std_logic;
```



```
DATA5      : in std_logic;
Conv_OK     : out std_logic;
CS          : out std_logic;
CLK_ADC     : out std_logic;
DATO1       : out std_logic_vector (9 downto 0);
DATO2       : out std_logic_vector (9 downto 0);
DATO3       : out std_logic_vector (9 downto 0);
DATO4       : out std_logic_vector (9 downto 0);
DATO5       : out std_logic_vector (9 downto 0));
```

end AD_serial_converter;

architecture Behavioral of AD_serial_converter is

```
type estados is (espera,captura);
signal Estado      : estados;
signal contador    : integer range 0 to 1; -- 1 bit <- Para el reloj
signal conta      : integer range 0 to 31; -- 5 bits <-Para la conversion
signal SCLK       : std_logic;
signal Chip_Select : std_logic;
signal Fin        : std_logic;
signal SDO1       : std_logic;
signal SDO2       : std_logic;
signal SDO3       : std_logic;
signal SDO4       : std_logic;
signal SDO5       : std_logic;
signal Registro1   : std_logic_vector (9 downto 0);
signal Registro2   : std_logic_vector (9 downto 0);
signal Registro3   : std_logic_vector (9 downto 0);
signal Registro4   : std_logic_vector (9 downto 0);
signal Registro5   : std_logic_vector (9 downto 0);
```

begin

CLK_ADS_PROC : process(clk, Reset) is

```
begin
    if (Reset='1') then
        contador <= 0;
        SCLK      <= '0';
    elsif (clk'event and clk='1') then
        if (contador = 1) then -- Conversión de frecuencia 50MHz ->
                                -- 6.25MHz
            SCLK      <= not SCLK;
            contador <= 0;
        else
            contador <= contador + 1;
        end if;
    end if;
```



```
        end if;

        end process CLK_ADS_PROC ;

--SEL_CAPTADOR_PROC: process(SCLK,Reset) is
--    begin
--
--        if (Reset='1') then
--            SDO <= '0';
--
--        elsif SCLK'event and SCLK='0' then
--            case Switch is
--                when "10000"=>
--                    SDO1 <= DATA1;
--                when "01000" =>
--                    SDO2 <= DATA2;
--                when "00100"=>
--                    SDO3 <= DATA3;
--                when "00010" =>
--                    SDO4 <= DATA4;
--                when "00001" =>
--                    SDO5 <= DATA5;
--                when others =>
--
--            end case;
--        end if;
--    end process SEL_CAPTADOR_PROC ;

CHIP_SEL_PROC : process(SCLK,Reset,Conv_ST) is
    begin

        if (Reset='1') then
            Chip_Select <= '1';
        elsif SCLK'event and SCLK='0'    then
            case estado is
                when captura =>
                    if (conta=1) then
                        Chip_Select <= '1';
                    else
                        Chip_Select <= '0';
                    end if;
                when others =>
                    if (Conv_ST = '1') then
                        Chip_Select <= '0';
                    else
                        Chip_Select <= '1';
                    end if;
            end case;
        end case;
```



```
        end if;
    end process CHIP_SEL_PROC;

CONTADOR_ADS : process (SCLK, Reset) is
begin
    if (Reset = '1') then
        conta <= 16;
    elsif SCLK'event and SCLK = '0' then
        case estado is
            when espera =>
                conta <= 16;
            when captura =>
                if (conta = 0) then
                    conta <= 16;
                else
                    conta <= conta - 1;
                end if;
            when others =>
            end case;
        end if;

    end process CONTADOR_ADS;

Registro_Estados : process(Chip_Select, Reset, Conv_ST) is
begin
    if (Reset = '1') then
        Estado <= espera;
    elsif Conv_ST='0' then
        Estado <= espera;
    elsif Chip_Select'event and Chip_Select='0' then
        Estado <= captura;
    end if;
end process Registro_Estados;

DATA_OUTPUT : process (SCLK, Reset) is
begin
    if Reset = '1' then
        Fin <= '0';
    elsif SCLK'event and SCLK='0' then
        case estado is
            when captura =>
                if (conta <= 12) then
                    if (conta >= 3) then
                        Registro1(conta-3) <= SDO1;
                        Registro2(conta-3) <= SDO2;
```



```

                                Registro3(conta-3) <= SDO3;
                                Registro4(conta-3) <= SDO4;
                                Registro5(conta-3) <= SDO5;
                                Fin <= '0';
                                end if;
                                end if;
                                if (conta = 1) then
                                    Fin <= '1';
                                    DATO1 <= Registro1;
                                    DATO2 <= Registro2;
                                    DATO3 <= Registro3;
                                    DATO4 <= Registro4;
                                    DATO5 <= Registro5;
                                else
                                    Fin <= '0';
                                end if;
                                when others =>
                                    Fin <= '0';
                                end case;
                                end if;
                                end process DATA_OUTPUT;

                                CLK_ADC <= SCLK;
                                CS    <= Chip_Select;
                                Conv_OK <= Fin;

                                end Behavioral;
```



CÓDIGO SIMULABLE

- BANCO DE PRUEBAS: *TESTBENCH.VHD*

```
-----
-- Company: UC3M
-- Engineer: Leandro E. Boyano Vannucchi
--
-- Create Date: 16:01:50 07/17/2007
-- Design Name: AD_serial_converter
-- Module Name:
D:/LEANDRO/UNISOL/ADC_serie/VHDL/ADserial/testbench.vhd
-- Project Name: ADserial
-- Target Device:
-- Tool versions:
-- Description:
--
-- VHDL Test Bench Created by ISE for module: AD_serial_converter
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
-----

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_unsigned.all;
USE ieee.numeric_std.ALL;

ENTITY testbench_vhd IS
END testbench_vhd;

ARCHITECTURE behavior OF testbench_vhd IS
    constant cycle : time := 20 ns;
    -- Component Declaration for the Unit Under Test (UUT)
    COMPONENT AD_serial_converter
    PORT(
        Conv_ST : IN std_logic;
        clk      : IN std_logic;
        Reset    : IN std_logic;
        DATA1   : IN std_logic;
        DATA2   : IN std_logic;
        DATA3   : IN std_logic;
        DATA4   : IN std_logic;
```

```
DATA5 : IN std_logic;
Conv_OK : OUT std_logic;
CS      : OUT std_logic;
CLK_ADC : OUT std_logic;
DATO1   : OUT std_logic_vector(9 downto 0);
DATO2   : OUT std_logic_vector(9 downto 0);
DATO3   : OUT std_logic_vector(9 downto 0);
DATO4   : OUT std_logic_vector(9 downto 0);
DATO5   : OUT std_logic_vector(9 downto 0);
);
END COMPONENT;
```

--Inputs

```
SIGNAL Conv_ST : std_logic := '0';
SIGNAL clk : std_logic := '0';
SIGNAL Reset : std_logic := '0';
SIGNAL SDO1 : std_logic := '0';
SIGNAL SDO2 : std_logic := '0';
SIGNAL SDO3 : std_logic := '0';
SIGNAL SDO4 : std_logic := '0';
SIGNAL SDO5 : std_logic := '0';
```

--Outputs

```
SIGNAL Conv_OK : std_logic;
SIGNAL CS : std_logic;
SIGNAL CLK_ADC : std_logic;
SIGNAL DATO1 : std_logic_vector(9 downto 0);
SIGNAL DATO2 : std_logic_vector(9 downto 0);
SIGNAL DATO3 : std_logic_vector(9 downto 0);
SIGNAL DATO4 : std_logic_vector(9 downto 0);
SIGNAL DATO5 : std_logic_vector(9 downto 0);
```

BEGIN

-- Instantiate the Unit Under Test (UUT)

```
uut: AD_serial_converter PORT MAP(
    Conv_ST => Conv_ST,
    clk => clk,
    Reset => Reset,
    DATA1 => SDO1,
    DATA2 => SDO2,
    DATA3 => SDO3,
    DATA4 => SDO4,
    DATA5 => SDO5,
    Conv_OK => Conv_OK,
    CS => CS,
    CLK_ADC => CLK_ADC,
    DATO1 => DATO1,
```



```
        DATO2 => DATO2,  
        DATO3 => DATO3,  
        DATO4 => DATO4,  
        DATO5 => DATO5  
    );
```

```
Clokgenerator : process  
begin  
    clk <= '1';  
    wait for cycle/2;  
    clk <= '0';  
    wait for cycle/2;  
end process Clokgenerator;
```

```
Principal : process  
begin  
    Reset <= '1';  
    wait for 100*cycle;  
    Reset <= '0';  
    wait for 70000*cycle;  
    assert false  
        report "End of simulation"  
        severity failure;  
end process Principal;
```

```
DATOS : PROCESS  
BEGIN
```

```
    SDO1 <= '0';  
    SDO2 <= '0';  
    SDO3 <= '0';  
    SDO4 <= '0';  
    SDO5 <= '0';  
    wait for 253*cycle;  
    SDO1 <= '0';  
    SDO2 <= '0';  
    SDO3 <= '0';  
    SDO4 <= '0';  
    SDO5 <= '0';  
    wait for 4*cycle;  
    SDO1 <= '0';  
    SDO2 <= '0';  
    SDO3 <= '0';  
    SDO4 <= '0';  
    SDO5 <= '0';  
    wait for 4*cycle;  
    SDO1 <= '0';
```




```
SDO2 <= '0';
SDO3 <= '0';
SDO4 <= '0';
SDO5 <= '0';
wait for 4*cycle;
SDO1 <= '0';
SDO2 <= '0';
SDO3 <= '0';
SDO4 <= '0';
SDO5 <= '0';
wait for 4*cycle;
SDO1 <= '1';
SDO2 <= '1';
SDO3 <= '1';
SDO2 <= '1';
SDO3 <= '1';
wait for 4*cycle;
SDO1 <= '1';
SDO2 <= '1';
SDO3 <= '1';
SDO4 <= '0';
SDO5 <= '0';
wait for 4*cycle;
SDO1 <= '0';
SDO2 <= '0';
SDO3 <= '0';
SDO4 <= '0';
SDO5 <= '0';
wait for 4*cycle;
SDO1 <= '1';
SDO2 <= '1';
SDO3 <= '1';
SDO4 <= '1';
SDO5 <= '1';
wait for 4*cycle;
SDO1 <= '1';
SDO2 <= '1';
SDO3 <= '1';
SDO4 <= '1';
SDO5 <= '1';
wait for 4*cycle;
SDO1 <= '0';
SDO2 <= '0';
SDO3 <= '0';
SDO4 <= '0';
SDO5 <= '0';
wait for 4*cycle;
```

```
SDO1 <= '0';
SDO2 <= '0';
SDO3 <= '0';
SDO4 <= '0';
SDO5 <= '0';
wait for 4*cycle;
SDO1 <= '1';
SDO2 <= '1';
SDO3 <= '1';
SDO4 <= '1';
SDO5 <= '1';
wait for 4*cycle;
SDO1 <= '0';
SDO2 <= '0';
SDO3 <= '0';
SDO4 <= '0';
SDO5 <= '0';
wait for 4*cycle;
SDO1 <= '1';
SDO2 <= '1';
SDO3 <= '1';
SDO4 <= '1';
SDO5 <= '1';
wait for 4*cycle;
SDO1 <= '0';
SDO2 <= '0';
SDO3 <= '0';
SDO4 <= '0';
SDO5 <= '0';
wait for 4*cycle;
SDO1 <= '0';
SDO2 <= '0';
SDO3 <= '0';
SDO4 <= '0';
SDO5 <= '0';
```

--Siguiente conversión tras hacer el ChipSelect

```
wait for 8*cycle;
SDO1 <= '0';
SDO2 <= '0';
SDO3 <= '0';
SDO4 <= '0';
SDO5 <= '0';
wait for 4*cycle;
SDO1 <= '0';
SDO2 <= '0';
SDO3 <= '0';
```

```
SDO4 <= '1';
SDO5 <= '1';
wait for 4*cycle;
SDO1 <= '0';
SDO2 <= '0';
SDO3 <= '0';
SDO4 <= '0';
SDO5 <= '0';
wait for 4*cycle;
SDO1 <= '0';
SDO2 <= '0';
SDO3 <= '0';
SDO4 <= '0';
SDO5 <= '0';
wait for 4*cycle;
SDO1 <= '1';
SDO2 <= '1';
SDO3 <= '1';
SDO4 <= '1';
SDO5 <= '1';
wait for 4*cycle;
SDO1 <= '1';
SDO2 <= '1';
SDO3 <= '1';
SDO4 <= '1';
SDO5 <= '1';
wait for 4*cycle;
SDO1 <= '0';
SDO2 <= '0';
SDO3 <= '0';
SDO4 <= '0';
SDO5 <= '0';
wait for 4*cycle;
SDO1 <= '0';
SDO2 <= '0';
SDO3 <= '0';
SDO4 <= '0';
SDO5 <= '0';
wait for 4*cycle;
SDO1 <= '1';
SDO2 <= '1';
SDO3 <= '1';
SDO4 <= '1';
SDO5 <= '1';
wait for 4*cycle;
SDO1 <= '0';
SDO2 <= '0';
```

```
SDO3 <= '0';
SDO4 <= '0';
SDO5 <= '0';
wait for 4*cycle;
SDO1 <= '1';
SDO2 <= '1';
SDO3 <= '1';
SDO4 <= '1';
SDO5 <= '1';
wait for 4*cycle;
SDO1 <= '1';
SDO2 <= '1';
SDO3 <= '1';
SDO4 <= '1';
SDO5 <= '1';
wait for 4*cycle;
SDO1 <= '1';
SDO2 <= '1';
SDO3 <= '1';
SDO4 <= '1';
SDO5 <= '1';
wait for 4*cycle;
SDO1 <= '1';
SDO2 <= '1';
SDO3 <= '1';
SDO4 <= '1';
SDO5 <= '1';
wait for 4*cycle;
SDO1 <= '0';
SDO2 <= '0';
SDO3 <= '0';
SDO4 <= '0';
SDO5 <= '0';
wait for 4*cycle;
SDO1 <= '0';
SDO2 <= '0';
SDO3 <= '0';
SDO4 <= '0';
SDO5 <= '0';
wait for 4*cycle;
```

--Siguiente conversión tras hacer el ChipSelect

```
SDO1 <= '0';
SDO2 <= '0';
SDO3 <= '0';
SDO4 <= '0';
```



```
SDO5 <= '0';
wait for 16*cycle;
SDO1 <= '0';
SDO2 <= '0';
SDO3 <= '0';
SDO4 <= '0';
SDO5 <= '0';
wait for 4*cycle;
SDO1 <= '1';
SDO2 <= '1';
SDO3 <= '1';
SDO4 <= '1';
SDO5 <= '1';
wait for 4*cycle;
SDO1 <= '0';
SDO2 <= '0';
SDO3 <= '0';
SDO4 <= '0';
SDO5 <= '0';
wait for 4*cycle;
SDO1 <= '1';
SDO2 <= '1';
SDO3 <= '1';
SDO4 <= '1';
SDO5 <= '1';
wait for 4*cycle;
SDO1 <= '0';
SDO2 <= '0';
SDO3 <= '0';
SDO4 <= '0';
SDO5 <= '0';
wait for 4*cycle;
SDO1 <= '1';
SDO2 <= '1';
SDO3 <= '1';
SDO4 <= '1';
SDO5 <= '1';
wait for 4*cycle;
SDO1 <= '1';
SDO2 <= '1';
SDO3 <= '1';
SDO4 <= '1';
SDO5 <= '1';
wait for 4*cycle;
SDO1 <= '1';
SDO2 <= '1';
SDO3 <= '1';
```

```
SDO4 <= '1';
SDO5 <= '1';
wait for 4*cycle;
SDO1 <= '0';
SDO2 <= '0';
SDO3 <= '0';
SDO4 <= '0';
SDO5 <= '0';
wait for 4*cycle;
SDO1 <= '1';
SDO2 <= '0';
SDO3 <= '0';
SDO4 <= '0';
SDO5 <= '0';
wait for 4*cycle;
SDO1 <= '0';
SDO2 <= '0';
SDO3 <= '0';
SDO4 <= '0';
SDO5 <= '0';
wait for 4*cycle;
SDO1 <= '0';
SDO2 <= '0';
SDO3 <= '0';
SDO4 <= '0';
SDO5 <= '0';
wait for 4*cycle;

--Siguiente conversión tras hacer el ChipSelect
SDO1 <= '0';
SDO2 <= '0';
SDO3 <= '0';
SDO4 <= '0';
SDO5 <= '0';
wait for 16*cycle;
SDO1 <= '0';
SDO2 <= '0';
SDO3 <= '0';
SDO4 <= '0';
SDO5 <= '0';
wait for 4*cycle;
SDO1 <= '1';
SDO2 <= '1';
SDO3 <= '1';
SDO4 <= '1';
SDO5 <= '1';
wait for 4*cycle;
```

```
SDO1 <= '0';
SDO2 <= '0';
SDO3 <= '0';
SDO4 <= '0';
SDO5 <= '0';
wait for 4*cycle;
SDO1 <= '1';
SDO2 <= '1';
SDO3 <= '1';
SDO4 <= '1';
SDO5 <= '1';
wait for 4*cycle;
SDO1 <= '0';
SDO2 <= '0';
SDO3 <= '0';
SDO4 <= '0';
SDO5 <= '0';
wait for 4*cycle;
SDO1 <= '1';
SDO2 <= '1';
SDO3 <= '1';
SDO4 <= '1';
SDO5 <= '1';
wait for 4*cycle;
SDO1 <= '1';
SDO2 <= '1';
SDO3 <= '1';
SDO4 <= '1';
SDO5 <= '1';
wait for 4*cycle;
SDO1 <= '0';
SDO2 <= '0';
SDO3 <= '0';
SDO4 <= '0';
SDO5 <= '0';
wait for 4*cycle;
SDO1 <= '1';
SDO2 <= '1';
SDO3 <= '1';
SDO4 <= '1';
SDO5 <= '1';
```



```
wait for 4*cycle;
SDO1 <= '1';
SDO2 <= '1';
SDO3 <= '1';
SDO4 <= '1';
SDO5 <= '1';
wait for 4*cycle;
SDO1 <= '0';
SDO2 <= '0';
SDO3 <= '0';
SDO4 <= '0';
SDO5 <= '0';
wait for 35*cycle;

END PROCESS;

CONVERSION : PROCESS
BEGIN

    Conv_ST <= '0';
    wait for 5.015 us;
    Conv_ST <= '1';
    wait; -- will wait forever
END PROCESS;

END;
```




ANEXO VIII. DESCRIPCIÓN VHDL PARA EL CONTROL DEL REGULADOR POR HISTÉRESIS

CÓDIGO SINTETIZABLE Y SIMULABLE

- *ADC_Y_LAZOI_AJUSTABLE.VHD*

```
-----
-- Company: UC3M
-- Engineer: Leandro E. Boyano Vannucchi
--
-- Create Date: 12:50:09 07/24/2008
-- Design Name:
-- Module Name: ADC_y_LazoI- Behavioral
-- Project Name:
-- Target Devices:
-- Tool versions:
-- Description:
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
-----

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
use IEEE.NUMERIC_STD.ALL;

---- Uncomment the following library declaration if instantiating
---- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity ADC_y_LazoI is
Port (
    Switch1 : in std_logic_vector(0 to 1); -- Selector de ajuste de
offset de tensión medida por el sensor
    Switch2 : in std_logic_vector(0 to 1); -- Selector de ajuste de
offset de corriente medida por el sensor
    Switch3 : in std_logic; -- Selector de ajuste de los límites para el
```

```

regulador por histéresis
    Switch4      : in std_logic; -- Selector de la Ganancia para la
corriente de referencia
    Clk          : in std_logic;
    Conv_ST      : in std_logic;
    Reset        : in std_logic;
    Io_serie     : in std_logic;
    Ured_serie   : in std_logic;
    Conv_Ok      : out std_logic;
    CS           : out std_logic;
    clk_ADC      : out std_logic;
    ADC_paral    : out std_logic_vector(9 downto 0); --Salida provisional
para probar la conversión de datos en serie a paralelo
    PWM1         : out std_logic;
    PWM2         : out std_logic;
    PWM3         : out std_logic;
    PWM4         : out std_logic;
    Io_PWM       : out std_logic; -- Salida para la reconstrucción de la
corriente tras la conversión de datos serie-paralelo. Señal únicamente de medición
    Ured_PWM     : out std_logic -- Salida para la reconstrucción de la
tensión de red tras la conversión de datos serie-paralelo. Señal únicamente de medición
);

end ADC_y_LazoI;

architecture Behavioral of ADC_y_LazoI is

    COMPONENT Control_Lazo_Cerrado
    PORT(
        Sel_Off_Ured : in std_logic_vector(0 to 1);
        Sel_Off_Io   : in std_logic_vector(0 to 1);
        Sel_Bounds_Iref : in std_logic;
        Sel_Gain_Iref : in std_logic;
        Clk          : in std_logic;
        Reset        : in std_logic;
        Io_digit     : in STD_LOGIC_VECTOR(9 downto 0);
        Ured_digit   : in STD_LOGIC_VECTOR(9 downto 0);
        PWM1         : out STD_LOGIC;
        PWM2         : out STD_LOGIC;
        PWM3         : out STD_LOGIC;
        PWM4         : out STD_LOGIC;
        clk_AD       : in std_logic;
        Io_sin_o     : out std_logic_vector(9 downto 0); -- Salida
únicamente para medición
        Ured_sin_o   : out std_logic_vector(9 downto 0) -- Salida
únicamente para medición
    );

```



```
END COMPONENT Control_Lazo_Cerrado;

COMPONENT AD_serial_converter
PORT(
    Conv_ST      : in std_logic;
    clk          : in std_logic;
    Reset        : in std_logic;
    DATA1       : in std_logic;
    DATA2       : in std_logic;
    Conv_OK      : out std_logic;
    CS           : out std_logic;
    CLK_ADC      : out std_logic;
    DATO1        : out std_logic_vector (9 downto 0);
    DATO2        : out std_logic_vector (9 downto 0);
);
END COMPONENT AD_serial_converter;

COMPONENT Digital_Analog_Converter
PORT(
    clk          : in std_logic;
    Reset        : in std_logic;
    ARRAY1       : in std_logic_vector (9 downto 0);
    ARRAY2       : in std_logic_vector (9 downto 0);
    pwm_DAC1     : out std_logic; -- Señal PWM de salida del
convertidor Digital-Analógico 1
    pwm_DAC2     : out std_logic -- Señal PWM de salida del
convertidor Digital-Analógico 2
);
END COMPONENT Digital_Analog_Converter;

-- Inputs

SIGNAL Clk_s      : std_logic;
SIGNAL Reset_s    : std_logic;
SIGNAL Conv_ST_s  : std_logic;
SIGNAL Io_serie_s : std_logic;
SIGNAL Ured_serie_s : std_logic;

-- Outputs

SIGNAL Chip_Select : std_logic; -- Señal que indica la frecuencia de
conversión de datos serie->paralelo
SIGNAL Led_convers : std_logic;
SIGNAL Clock_ADC   : std_logic;
SIGNAL PWM1_s      : std_logic;
SIGNAL PWM2_s      : std_logic;
SIGNAL PWM3_s      : std_logic;
```



```
SIGNAL PWM4_s : std_logic;

-- ADC serie-parallel converter to close loop control

SIGNAL Io_digital : std_logic_vector (9 downto 0);
SIGNAL Ured_digital : std_logic_vector (9 downto 0);
SIGNAL Io_so : std_logic_vector (9 downto 0);    --Señal únicamente para
medición
SIGNAL Ured_so : std_logic_vector (9 downto 0);    --Señal únicamente
para medición

-- Digital_Analog Outputs

SIGNAL Io_PWM_s : std_logic;
SIGNAL Ured_PWM_s : std_logic;

-- Close Loop Converter

SIGNAL Offset_Ured_s: std_logic_vector(1 downto 0);
SIGNAL Offset_Io_s : std_logic_vector(1 downto 0);
SIGNAL Hist_Bound_s : std_logic;
SIGNAL Gain_Iref_s : std_logic;

begin

-- Asignación de las entradas del módulo completo a las señales correspondientes

clk_s      <= clk;
Conv_ST_s  <= Conv_ST;
Reset_s    <= Reset;
Io_serie_s <= Io_serie;
Ured_serie_s <= Ured_serie;
Offset_Ured_s <= Switch1;
Offset_Io_s <= Switch2;
Hist_Bound_s <= Switch3;
Gain_Iref_s <= Switch4;

ADC_converter: AD_serial_converter port map(

    Conv_ST => Conv_ST_s,
    Clk => Clk_s,
    Reset => Reset_s,
    DATA1 => Io_serie_s,
    DATA2 => Ured_serie_s,
    Conv_OK => Led_convers,
    CS => Chip_Select,
    CLK_ADC => Clock_ADC,
```

```
DATO1 => Io_digital,
DATO2 => Ured_digital
);

DAC_converter: Digital_Analog_Converter port map(

    clk => clk_s,
    Reset => Reset_s,
    ARRAY1 => Io_so,
    ARRAY2 => Ured_so,
    PWM_DAC1 => Io_PWM_s,
    PWM_DAC2 => Ured_PWM_s
);

Control_Regulador: Control_Lazo_Cerrado PORT MAP(

    Sel_Off_Ured => Offset_Ured_s,
    Sel_Off_Io => Offset_Io_s,
    Sel_Bounds_Iref => Hist_Bound_s,
    Sel_Gain_Iref => Gain_Iref_s,
    Clk => Clk_s,
    Reset => Reset_s,
    Io_digit => Io_digital,
    Ured_digit => Ured_digital,
    clk_AD => Chip_Select,
    PWM1 => PWM1_s,
    PWM2 => PWM2_s,
    PWM3 => PWM3_s,
    PWM4 => PWM4_s,
    Io_sin_o => Io_so,    --Para pruebas
    Ured_sin_o=> Ured_so    --Para pruebas
);

-- Asignación de señales a las salidas del módulo completo

Conv_OK <= Led_convers;
CS    <= Chip_Select;
clk_ADC <= Clock_ADC;
PWM1  <= PWM1_s;
PWM2  <= PWM2_s;
PWM3  <= PWM3_s;
PWM4  <= PWM4_s;
ADC_paral <= Io_digital; -- Salida de prueba, datos en paralelo
Io_PWM  <= Io_PWM_s;
Ured_PWM <= Ured_PWM_s;

end;
```



- *MODULO_ADC_SERIE_PARALELO.VHD*

```
-----
-- Company: UC3M
-- Engineer: Leandro E. Boyano Vannucchi
-- Create Date: 15:35:28 10/04/2007
-- Design Name:
-- Module Name: ADC_modulo - Behavioral
-- Project Name:
-- Target Devices:
-- Tool versions:
-- Description: Módulo de código sintetizable que convierte una señal digital en serie a
valores en paralelo.
-- Registra valores de 10 bits por cada flanco de bajada de Chip_Select
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
-----

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
use IEEE.NUMERIC_STD.ALL;

--library UNISIM;
--use UNISIM.VComponents.all;

entity AD_serial_converter is

    port(

        Conv_ST : in std_logic;
        clk      : in std_logic;
        Reset    : in std_logic;
        DATA1   : in std_logic;
        DATA2   : in std_logic;
        Conv_OK   : out std_logic;
        CS        : out std_logic;
        CLK_ADC   : out std_logic;
        DATO1     : out std_logic_vector (9 downto 0);
        DATO2     : out std_logic_vector (9 downto 0)
    );

end AD_serial_converter;
```



architecture Behavioral of AD_serial_converter is

```
type estados is (espera,captura);
signal Estado      : estados;
signal contador    : integer range 0 to 1; -- 1 bit <- Para el reloj
signal conta       : integer range 0 to 31; -- 5 bits <-Para la conversion
signal SCLK        : std_logic;
signal Chip_Select : std_logic;
signal Fin         : std_logic;
signal SDO1        : std_logic;
signal SDO2        : std_logic;
signal Registro1   : std_logic_vector (9 downto 0);
signal Registro2   : std_logic_vector (9 downto 0);
```

begin

```
SDO1 <= DATA1;
SDO2 <= DATA2;
```

CLK_ADS_PROC : process(clk, Reset) is

```
begin
    if (Reset='1') then
        contador <= 0;
        SCLK      <= '0';
    elsif (clk'event and clk='1') then
        if (contador = 1) then -- Conversión de frecuencia 50MHz ->
6.25MHz
            SCLK      <= not SCLK;
            contador <= 0;
        else
            contador <= contador + 1;
        end if;
    end if;
end process CLK_ADS_PROC ;
```

CHIP_SEL_PROC : process(SCLK,Reset,Conv_ST) is

```
begin
    if (Reset='1') then
        Chip_Select <= '1';
    elsif SCLK'event and SCLK='0' then
        case estado is
            when captura =>
                if (conta=1) then
                    Chip_Select <= '1';
```



```

else
    Chip_Select <= '0';
end if;
when others =>
    if (Conv_ST = '1') then
        Chip_Select <= '0';
    else
        Chip_Select <= '1';
    end if;
end case;
end if;
end process CHIP_SEL_PROC;

CONTADOR_ADS : process (SCLK, Reset) is
begin
    if (Reset = '1') then
        conta <= 16;
    elsif SCLK'event and SCLK = '0' then
        case estado is
            when espera =>
                conta <= 16;
            when captura =>
                if (conta = 0) then
                    conta <= 16;
                else
                    conta <= conta - 1;
                end if;
            when others =>
                end case;
        end case;
    end if;
end process CONTADOR_ADS;

Registro_Estados : process(Chip_Select, Reset, Conv_ST) is
begin
    if (Reset = '1') then
        Estado <= espera;
    elsif Conv_ST='0' then
        Estado <= espera;
    elsif Chip_Select'event and Chip_Select='0' then
        Estado <= captura;
    end if;
end process Registro_Estados;

DATA_OUTPUT : process (SCLK, Reset) is
begin
```




```
        if Reset = '1' then
            Fin <= '0';
        elsif SCLK'event and SCLK='0' then
            case estado is
                when captura =>
                    if (conta <=12) then
                        if (conta >=3) then
                            Registro1(conta-3) <= SDO1;
                            Registro2(conta-3) <= SDO2;
                            Fin <= '0';
                        end if;
                    end if;
                    if (conta = 1) then
                        Fin <= '1';
                        DATO1 <= Registro1;
                        DATO2 <= Registro2;
                    else
                        Fin <= '0';
                    end if;
                when others =>
                    Fin <= '0';
            end case;
        end if;
    end process    DATA_OUTPUT;

    CLK_ADC <= SCLK;
    CS    <= Chip_Select;
    Conv_OK <= Fin;

end Behavioral;
```

- *MODULO_DAC.VHD*

```
-----
-- Company: UC3M
-- Engineer: Leandro E. Boyano Vannucchi
--
-- Create Date: 15:28:37 03/04/2009
-- Design Name:
-- Module Name: Modulo_DAC - Behavioral
-- Project Name:
-- Target Devices:
-- Tool versions:
-- Description: Este modulo compara un array digital de 10 bits con una señal
--              triangular, generada mediante un contador, de
--              manera que la salida es una señal PWM cuyo ciclo de trabajo
```



```
-- es proporcional al valor de la señal analógica original de la que procede
-- el array de 10 bits. De este modo, y mediante un filtro paso bajo, se
-- obtiene un valor analógico correspondiente a la señal digital de 10 bits.
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
-----
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
use IEEE.NUMERIC_STD.ALL;

entity Digital_Analog_Converter is

    port(
        clk          : in std_logic;
        Reset        : in std_logic;
        ARRAY1        : in std_logic_vector (9 downto 0);
        ARRAY2        : in std_logic_vector (9 downto 0);
        pwm_DAC1      : out std_logic; -- Señal PWM de salida del conversor Digital-
        Analógico 1
        pwm_DAC2      : out std_logic -- Señal PWM de salida del conversor Digital-
        Analógico 2
    );

end Digital_Analog_Converter;

architecture Behavioral of Digital_Analog_Converter is

    signal conta_DAC          : integer range 0 to 1023; -- 10 bits <-Para
    la triangular
    signal Registro_DAC1      : std_logic_vector (9 downto 0);
    signal Registro_DAC2      : std_logic_vector (9 downto 0);
    signal pwm_out1           : std_logic;
    signal pwm_out2           : std_logic;

begin
    Registro_DAC1 <= ARRAY1+512; --Se suma 512 para centrar la señal a la
    mitad de la rampa y observar los resultados centrados.
    Registro_DAC2 <= ARRAY2+512; --Se suma 512 para centrar la señal a la
    mitad de la rampa y observar los resultados centrados.

    GENERACION_TRIANGULAR : process(clk, Reset) is
```



```
begin
    if (Reset='1') then
        conta_DAC <= 0;
    elsif (clk'event and clk='1') then
        conta_DAC <= conta_DAC+1;
    end if;
end process GENERACION_TRIANGULAR;

GENERACION_PWM : process(clk, Reset) is
begin
    if (Reset='1') then
        pwm_out1 <= '0';
        pwm_out2 <= '0';
    elsif (clk'event and clk='1') then
        if Registro_DAC1 >= conta_DAC then
            pwm_out1 <='1';
        else
            pwm_out1 <='0';
        end if;
        if Registro_DAC2 >= conta_DAC then
            pwm_out2 <='1';
        else
            pwm_out2 <='0';
        end if;
    end if;
end process GENERACION_PWM;

pwm_DAC1 <= pwm_out1;
pwm_DAC2 <= pwm_out2;

end Behavioral;
```

- *CONTROL_LAZO_CERRADO_AJUSTABLE.VHD*

```
-----
-- Company: UC3M
-- Engineer: Leandro E. Boyano Vannucchi
--
-- Create Date: 12:50:09 07/24/2008
-- Design Name:
-- Module Name: Control_Lazo_Cerrado - Behavioral
-- Project Name:
-- Target Devices:
-- Tool versions:
-- Description: En este módulo se adapta el offset de las señales de tensión y corriente
```



```
--      para posteriormente generar una referencia de corriente con la señal de la
--      tensión y crear unos límites que van a servir para hacer un regulador por
--      histeresis.
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
-----
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
use IEEE.NUMERIC_STD.ALL;

entity Control_Lazo_Cerrado is

-- Definición del offset que hay que quitar a los valores de Io y Ured digitalizados
-- Definición de constantes propias del regulador

generic(
    G1: integer :=2; -- CONSTANTE DE TRANSFORMACION
    DE TENSION A CORRIENTE. Equivale a Iref=0'5A
    G2: integer :=4; -- Equivale a Iref=0'25A
    t_muerto: integer :=30 -- Tiempo muerto para transistores
    -- complementarios
); -- Ejemplo: Para 1us de t. muerto serán 50ciclos*20ns,
-- siendo el reloj de la FPGA de f=50MHz(T=20ns)

Port (
    Sel_Off_Ured : in std_logic_vector(0 to 1):= (others =>'0'); -- Para
eliminar el offset de la señal medida por el sensor de tension
    Sel_Off_Io : in std_logic_vector(0 to 1):= (others =>'0'); --
Para eliminar el offset de la señal medida por el sensor de corriente
    Sel_Bounds_Iref : in std_logic; --
Para ajustar los límites de comparación para la regulación por histéresis
    Sel_Gain_Iref : in std_logic;
    Clk : in std_logic;
    Reset : in std_logic;
    Io_digit : in STD_LOGIC_VECTOR(9 downto 0):= (others =>'0');
    Ured_digit : in STD_LOGIC_VECTOR(9 downto 0):= (others
=>'0');

    PWM1 : out STD_LOGIC;
    PWM2 : out STD_LOGIC;
    PWM3 : out STD_LOGIC;
    PWM4 : out STD_LOGIC;
```



```
        clk_AD      : in std_logic;  -- Señal a la frecuencia de conversión del
ADC -> 735.294KHz
        Io_sin_o     : out std_logic_vector(9 downto 0):= (others =>'0'); --
Solo para medición
        Ured_sin_o    : out std_logic_vector(9 downto 0):= (others =>'0')
        --Solo para medición
    );

end Control_Lazo_Cerrado;

architecture Behavioral of Control_Lazo_Cerrado is

    SIGNAL Io_int_sin_o    : integer;
    SIGNAL Ured_int_sin_o  : integer;

    SIGNAL REF_pos         : integer;
    SIGNAL REF_neg         : integer;
    SIGNAL PWM             : std_logic:='1';
    SIGNAL PWM_aux         : std_logic;
    SIGNAL conta_tm        : integer :=0;
    SIGNAL tm              : std_logic:='0';
    signal ena_ADC         : std_logic;

    -- SEÑALES PARA EL AJUSTE MANUAL DE LOS PARÁMETROS DEL LAZO
    DE CONTROL

    SIGNAL ref_offset_tension : integer;
    SIGNAL ref_offset_corriente : integer;
    SIGNAL Limites_histeresis : integer;
    SIGNAL REF_corriente      : integer;

begin

Enable_ADC: process (clk,Reset) is
begin
    if Reset='1' then
        ena_ADC<='0';
    elsif clk='1' and clk'event then
        if ena_ADC='0' and Clk_AD='0' then
            ena_ADC<='0';
        elsif ena_ADC='0' and Clk_AD='1' then
            ena_ADC<='1';
        elsif ena_ADC='1' then
            ena_ADC<='0';
        end if;
    end if;
end process Enable_ADC;
```

-- Ajuste del offset a eliminar en la señal de la tensión de red sensorizada

SEL_OFFSET_Ured_PROC: process(clk,Reset) is

begin

if (Reset='1') then

ref_offset_tension <= 321; --Valor +/- estable

elsif clk'event and clk='1' and ena_ADC='1' then

case Sel_Off_Ured is

when "00"=>

ref_offset_tension <= 321;

when "01"=>

ref_offset_tension <= 311;

when "10"=>

ref_offset_tension <= 316;

when "11"=>

ref_offset_tension <= 326;

when others =>

end case;

end if;

end process SEL_OFFSET_Ured_PROC;

-- Ajuste del offset a eliminar en la señal de corriente de salida sensorizada

SEL_OFFSET_Io_PROC: process(clk,Reset) is

begin

if (Reset='1') then

ref_offset_corriente <= 187; --Valor +/- estable

elsif clk'event and clk='1' and ena_ADC='1' then

case Sel_Off_Io is

when "00"=>

ref_offset_corriente <= 187;

when "01"=>

ref_offset_corriente <= 194;

when "10"=>

ref_offset_corriente <= 197;

when "11"=>

ref_offset_corriente <= 199;

when others =>

end case;



```
        end if;
    end process SEL_OFFSET_Io_PROC;

    -- Ajuste de los límites de comparación para el regulador por histéresis

    SEL_Limit_Histeresis_PROC: process(clk,Reset) is

        begin
            if (Reset='1') then
                Limites_histeresis <= 8;
            elsif clk'event and clk='1' and ena_ADC='1' then
                case Sel_Bounds_Iref is
                    when '0'=>
                        Limites_histeresis <= 8;
                    when '1'=>
                        Limites_histeresis <= 32;
                    when others =>
                        --
                end case;
            end if;
        end process SEL_Limit_Histeresis_PROC;

    --Se elimina el offset de la señales digitales para centrarlas en el origen

    Elimina_offset: process(clk, Reset)

        begin
            if Reset='1' then
                Io_int_sin_o    <= ref_offset_corriente;
                Ured_int_sin_o  <= ref_offset_tension;
            elsif clk'event and clk='1' and ena_ADC='1' then
                Io_int_sin_o    <= CONV_INTEGER(Io_digit) -
ref_offset_corriente;
                Ured_int_sin_o  <= CONV_INTEGER(Ured_digit) -
ref_offset_tension;
            end if;
        end process Elimina_offset;

    --Cálculo de la corriente de referencia

    corriente_ref: process(clk, Reset)
        begin
            if Reset = '1' then
                REF_corriente <= 0;
            elsif clk'event and clk='1' and ena_ADC='1' then
                case Sel_Gain_Iref is
                    when '0'=>
                        REF_corriente <= Ured_int_sin_o/G1;
                end case;
            end if;
        end process corriente_ref;
```



```
when '1' =>
    REF_corriente <= Ured_int_sin_o/G2;
when others =>
    end case;
end if;
end process corriente_ref;

REF_pos <= REF_corriente + Limites_histeresis;
REF_neg <= REF_corriente - Limites_histeresis;

--Registro de desplazamiento para la señal de error (a la frecuencia de muestreo del
regulador (10kHz))

Regulador_con_histeresis: process(clk)
begin
    if Reset='1' then
        PWM<='0';
    elsif clk'event and clk='1' and ena_ADC='1' then
        if PWM = '1' and Io_int_sin_o < REF_pos then
            PWM <='1';
        elsif PWM = '1' and Io_int_sin_o > REF_pos then
            PWM <='0';
        elsif PWM = '0' and Io_int_sin_o > REF_neg then
            PWM <='0';
        elsif PWM = '0' and Io_int_sin_o < REF_neg then
            PWM <='1';
        end if;
    end if;
end process Regulador_con_histeresis;

-- Retraso de un ciclo para generar las PWM's sin que haya glitches

process (clk)
begin
    if Reset='1' then
        PWM_aux <='0';
    elsif clk'event and clk='1' then
        PWM_aux <= PWM;
    end if;
end process;

-- Contador para definir el tiempo muerto

Contador_tmuerto : process(clk, Reset) is
begin
    if Reset='1' then
```



```
        conta_tm <= 0;
    elsif clk'event and clk='1' then
        if PWM = '1' and Io_int_sin_o > REF_pos then
            conta_tm <= 1;
        elsif PWM = '1' and Io_int_sin_o < REF_pos then
            if conta_tm < t_muerto then
                conta_tm <= conta_tm + 1;
            else
                conta_tm <= t_muerto;
            end if;
        elsif PWM = '0' and Io_int_sin_o < REF_neg then
            conta_tm <= 1;
        elsif PWM = '0' and Io_int_sin_o > REF_neg then
            if conta_tm < t_muerto then
                conta_tm <= conta_tm + 1;
            else
                conta_tm <= t_muerto;
            end if;
        end if;
    end if;
end process Contador_tmuerto;

-- Generación de la señal de tiempo muerto

tmuerto: process (conta_tm)
begin
    if conta_tm < t_muerto then
        tm <= '1';
    else
        tm <= '0';
    end if;
end process tmuerto;

PWM1 <= PWM_aux and not tm;
PWM2 <= not PWM_aux and not tm;
PWM3 <= not PWM_aux and not tm;
PWM4 <= PWM_aux and not tm;
Io_sin_o<= CONV_STD_LOGIC_VECTOR(Io_int_sin_o,10);
Ured_sin_o<= CONV_STD_LOGIC_VECTOR(REF_corriente,10);
end Behavioral;
```



CÓDIGO SIMULABLE

- BANCO DE PRUEBAS: *BANCO_DE_PRUEBAS.VHD*

```
-----
-- Company: UC3M
-- Engineer: Leandro E. Boyano Vannucchi
--
-- Create Date: 10:57:01 07/28/2008
-- Design Name: Control_Lazo_Cerrado
-- Module Name: E:/LEANDRO/UNISOL/Regulador/Banco_de_pruebas.vhd
-- Project Name: Regulador
-- Target Device:
-- Tool versions:
-- Description:
--
-- VHDL Test Bench Created by ISE for module: Control_Lazo_Cerrado
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
-- Notes:
-- This testbench has been automatically generated using types std_logic and
-- std_logic_vector for the ports of the unit under test. Xilinx recommends
-- that these types always be used for the top-level I/O of a design in order
-- to guarantee that the testbench will bind correctly to the post-implementation
-- simulation model.
-----
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_unsigned.all;
use IEEE.STD_LOGIC_ARITH.ALL;
USE ieee.numeric_std.ALL;
use WORK.DWmath.all;
use WORK.MyconvPack.all;

ENTITY Banco_de_pruebas_vhd IS
END Banco_de_pruebas_vhd;

ARCHITECTURE behavior OF Banco_de_pruebas_vhd IS

    constant cycle_clk: time := 20 ns; -- Frecuencia de reloj: 50MHz

    -- Component Declaration for the Unit Under Test (UUT)
```

```

COMPONENT ADC_y_LazoI
PORT (
    Switch1      : in std_logic_vector(0 to 1);
    Switch2      : in std_logic_vector(0 to 1);
    Switch3      : in std_logic;
    Switch4      : in std_logic;
    Clk           : in std_logic;
    Conv_ST      : in std_logic;
    Reset        : in std_logic;
    Io_serie     : in std_logic;
    Ured_serie   : in std_logic;
    Conv_Ok      : out std_logic;
    CS           : out std_logic;
    clk_ADC      : out std_logic;
    --           ADC_paral : out std_logic_vector(9 downto 0); --Salida provisional
para probar la conversion la conversion de datos en serie a paralelo
    PWM1        : out std_logic;
    PWM2        : out std_logic;
    PWM3        : out std_logic;
    PWM4        : out std_logic;
    --           Io_so_paral : out std_logic_vector(9 downto 0)
    );
END COMPONENT;

COMPONENT Invertermodel
PORT(
    Ue          : in real :=0.0;
    Ured        : in real :=0.0;
    OnOff1SM    : in std_logic; -- On = '1', off = '0'
    OnOff2SM    : in std_logic; -- On = '1', off = '0'
    OnOff3SM    : in std_logic; -- On = '1', off = '0'
    OnOff4SM    : in std_logic; -- On = '1', off = '0'
    ResN        : in std_logic;
    Ie          : out real :=0.0;
    Io          : out real :=0.0;
    Uab         : out real :=0.0
    );
END COMPONENT;

COMPONENT Modelo_ADC
PORT(
    Clk          : in std_logic;
    Clk_ADC      : in std_logic;
    Reset        : in std_logic;
    Chip_Sel     : in std_logic;
    Io           : in real :=0.0;

```



```
        Ured      : in real :=0.0;
        Io_serie   : out std_logic;
        Ured_serie : out std_logic
    );
END COMPONENT;

--Inputs

SIGNAL Switch1 : std_logic_vector := "00";
SIGNAL Switch2 : std_logic_vector := "00";
SIGNAL Switch3 : std_logic := '0';
SIGNAL Switch4 : std_logic := '0';
SIGNAL Clk      : std_logic := '0';
SIGNAL Conv_ST  : std_logic := '0';
SIGNAL Reset    : std_logic := '0';
SIGNAL Io_tb    : real := 0.0;
SIGNAL Ured_tb  : real :=0.0;

--Outputs

SIGNAL PWM1 : std_logic;
SIGNAL PWM2 : std_logic;
SIGNAL PWM3 : std_logic;
SIGNAL PWM4 : std_logic;
SIGNAL clk_ADC : std_logic;

--Inverter model

signal Ue_tb    : real:=400.0;
signal Ie_tb    : real:=0.0;
signal Uinv_tb  : real:=0.0;

--ADC_y_LazoI

SIGNAL Io_s : STD_LOGIC;
SIGNAL Ured_s : STD_LOGIC;
SIGNAL Led_on : STD_LOGIC;
SIGNAL Chip_Select : STD_LOGIC;
-- SIGNAL Io_d : STD_LOGIC_VECTOR(9 downto 0);
-- SIGNAL Ured_d : STD_LOGIC_VECTOR(9 downto 0);

--Generador de senoidal

signal t      : real :=0.0;
constant dt   : real :=20.0e-9;

BEGIN
```

-- Instantiate the Unit Under Test (UUT)

```
uut: ADC_y_LazoI PORT MAP(  
    Switch1 => Switch1,  
    Switch2 => Switch2,  
    Switch3 => Switch3,  
    Switch4 => Switch4,  
    Clk => Clk,  
    Conv_ST => Conv_ST,  
    Reset => Reset,  
    Io_serie => Io_s,  
    Ured_serie => Ured_s,  
    Conv_OK => Led_on,  
    CS => Chip_Select,  
    clk_ADC => clk_ADC,  
    PWM1 => PWM1,  
    PWM2 => PWM2,  
    PWM3 => PWM3,  
    PWM4 => PWM4  
);
```

TheInverter_model: Invertermodel port map(
 Ue => Ue_tb,
 Ured => Ured_tb,
 OnOff1SM => PWM1,
 OnOff2SM => PWM2,
 OnOff3SM => PWM3,
 OnOff4SM => PWM4,
 ResN => Reset,
 Ie => Ie_tb,
 Io => Io_tb,
 Uab => Uinv_tb
);

ADC_converter: Modelo_ADC port map(
 Clk => Clk,
 clk_ADC => clk_ADC,
 Reset => Reset,
 Chip_Sel => Chip_Select,
 Io => Io_tb,
 Ured => Ured_tb,
 Io_serie => Io_s,
 Ured_serie => Ured_s
);

-- Generación de la señal de reloj de 50MHz



```
reloj:process
begin
    Clk <='0';
    wait for cycle_clk/2;
    Clk <='1';
    wait for cycle_clk/2;
end process reloj;

-- Se genera Ue utilizando la función "sin" de DWMath

ured_gen: process
begin
    t <= 0.0; -- Empieza al principio de ciclo
    loop
        wait for cycle_clk;
        t <= t + dt;
        Ured_tb <= 255.0*sin(2.0*pi*50.0*t);
    end loop;
end process ured_gen;

-- Se activa el conversor ADC de serie a paralelo

Activar_ADC: process
begin
    Conv_ST <='0';
    wait for cycle_clk*400;
    Conv_ST <='1';
    wait;
end process Activar_ADC;

tb : process
begin
    Reset <='1';
    wait for 6000 ns;
    Reset <='0';
    wait for cycle_clk/2;
    Reset <='1';
    wait for cycle_clk/2;
    Reset <='0';
    wait for cycle_clk/2;
    Reset <='1';
    wait for cycle_clk/2;
    Reset <='0';
    wait for cycle_clk/2;
    Reset <='1';
    wait for cycle_clk/2;
    Reset <='0';
```



```
wait for cycle_clk/2;
Reset <='1';
wait for cycle_clk/2;
Reset <='0';
wait for 1000 ms;
assert false
report "End of simulation"
severity failure;
end process tb;

END;
```

- MODELO DEL CONVERTIDOR ADC: *MODELO_ADC.VHD*

```
-----
-- Company: UC3M
-- Engineer: Leandro E. Boyano Vannucchi
--
-- Create Date: 12:50:09 07/24/2008
-- Design Name:
-- Module Name: Control_Lazo_Cerrado - Behavioral
-- Project Name:
-- Target Devices:
-- Tool versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
-----
```

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
use IEEE.NUMERIC_STD.ALL;
```

```
entity Modelo_ADC is
```

```
Port (
```

```
    Clk          : in std_logic;
    clk_ADC       : in std_logic;
    Reset         : in std_logic;
    Chip_Sel      : in std_logic;
    Io            : in real;
    Ured          : in real;
```



```
Io_serie    : out STD_LOGIC;
Ured_serie  : out STD_LOGIC
);
end Modelo_ADC;

architecture Behavioral of Modelo_ADC is

    SIGNAL Io_s          : integer:=0;
    SIGNAL Ured_s         : integer:=0;
    SIGNAL Io_dig         : STD_LOGIC_VECTOR(9 downto 0);
    SIGNAL Ured_dig        : STD_LOGIC_VECTOR(9 downto 0);
    SIGNAL conta_ADC      : integer:=0;
    SIGNAL Dato_Io_enviar  : STD_LOGIC_VECTOR(15 downto 0);
    SIGNAL Dato_Ured_enviar : STD_LOGIC_VECTOR(15 downto 0);
    SIGNAL Io_serie_aux    : STD_LOGIC;
    SIGNAL Ured_serie_aux  : STD_LOGIC;

begin

    -- Se convierte el valor real de Io y Ured a un valor entero entre 0 y 1024 para
    digitalizarlo con 10bits

    cambio_escala: process(Clk, Reset)

        begin
            if Reset='1' then
                Io_s  <= 0;
                Ured_s <= 0;
            elsif Clk'event and Clk='1' then
                if Io >= 8.01 then
                    Io_s <= 1023;
                elsif Io <= -5.99 then
                    Io_s <= 0;
                else
                    Io_s <= INTEGER(Io*53.0 + 438.0);
                end if;
                Ured_s <= INTEGER(Ured*1.95 + 524.3);
            end if;
        end process cambio_escala;

    -- Se convierte el valor entero de Io y Ured a un valor digital de 10bits, simulando el
    -- comportamiento de un ADC

    conversion_real_a_logic_vect: process(Clk_ADC, Reset)

        begin
            if Reset='1' then
```




```
        Io_dig  <= "000000000000";
        Ured_dig <= "000000000000";
    elsif Clk_ADC'event and Clk_ADC='1' then
        Io_dig  <= std_logic_vector(to_unsigned(Io_s,10));
        Ured_dig <= std_logic_vector(to_unsigned(Ured_s,10));
    end if;
end process conversion_real_a_logic_vect;

-- Paquete de datos a enviar en serie

Dato_Io_enviar(15 downto 12) <= (others =>'0');
Dato_Io_enviar(11 downto 2)  <= Io_dig;
Dato_Io_enviar(1 downto 0)  <= (others =>'0');
Dato_Ured_enviar(15 downto 12) <= (others =>'0');
Dato_Ured_enviar(11 downto 2) <= Ured_dig;
Dato_Ured_enviar(1 downto 0) <= (others =>'0');

-- Contador para el envío de bits de cada paquete de datos

Conta_bits: process(Clk_ADC, Reset, Chip_Sel)
begin
    if Reset='1' then
        conta_ADC <= 15;
    elsif Clk_ADC'event and Clk_ADC='0' then
        if Chip_Sel='0' then
            if conta_ADC= 0 then
                conta_ADC <= 15;
            else
                conta_ADC <= conta_ADC-1;
            end if;
        end if;
    end if;
end process Conta_bits;

Io_serie_aux  <= Dato_Io_enviar(conta_ADC);
Ured_serie_aux <= Dato_Ured_enviar(conta_ADC);

process(clk)
begin
    if clk'event and clk='1' then
        Io_serie  <= Io_serie_aux;
        Ured_serie <= Ured_serie_aux;
    end if;
end process;

end;
```



- MODELO DEL PUENTE INVERSOR: *INVERTER_MODEL.VHD*

Ver código simulable del puente inversor en el *ANEXO VI*.



ANEXO IX. ENSAYOS EXPERIMENTALES DEL INVERSOR EN LAZO CERRADO

Los ensayos realizados con el inversor en lazo cerrado se subdividen en 3 tipos de ensayo:

1. En cortocircuito.
2. Con carga resistiva.
3. Conexión a red.

Todas estas pruebas se han realizado con 2 referencias de corriente, denominadas **R1** y **R2**, cuyos valores de corriente de pico son: $V_{R1} = 1 A_{PICO}$ y $V_{R2} = 2 A_{PICO}$ respectivamente. Asimismo, se han empleado 2 bandas de histéresis diferentes, denominadas **H1** y **H2**, siendo H1 la banda de histéresis más estrecha y H2 la banda de histéresis más ancha.

La tensión máxima de entrada o tensión de panel, se ha limitado en estos ensayos a un máximo de 315 V. En todos

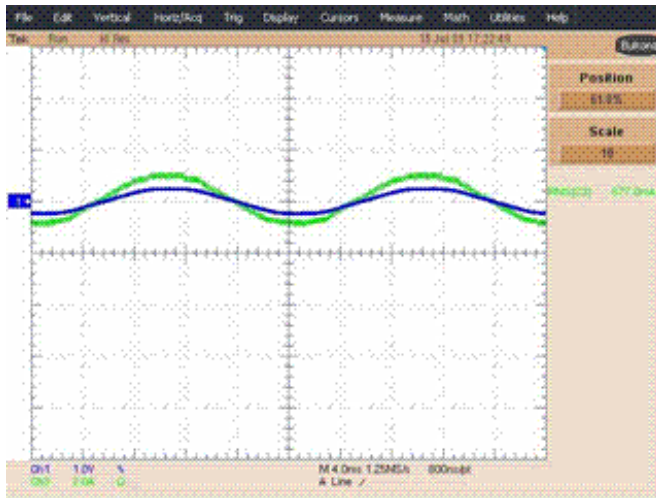
A. ENSAYOS EN CORTOCIRCUITO

Estas pruebas se han realizado empleando a la salida del inversor únicamente la bobina de conexión a red $L_2 = 8 \text{ mH}$, cuya impedancia a 50 Hz es de 2.56Ω .

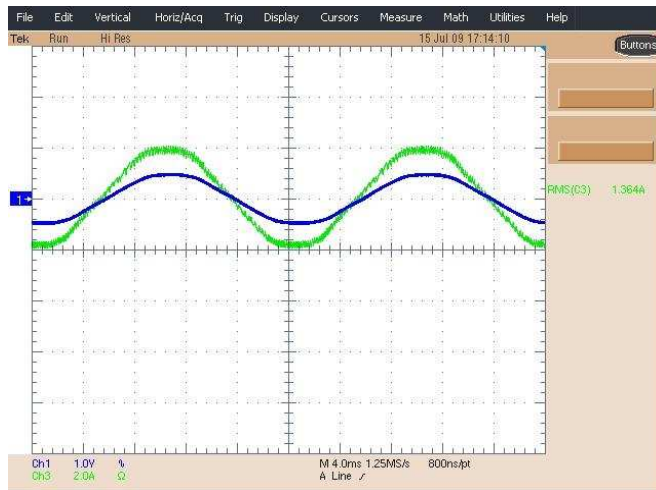
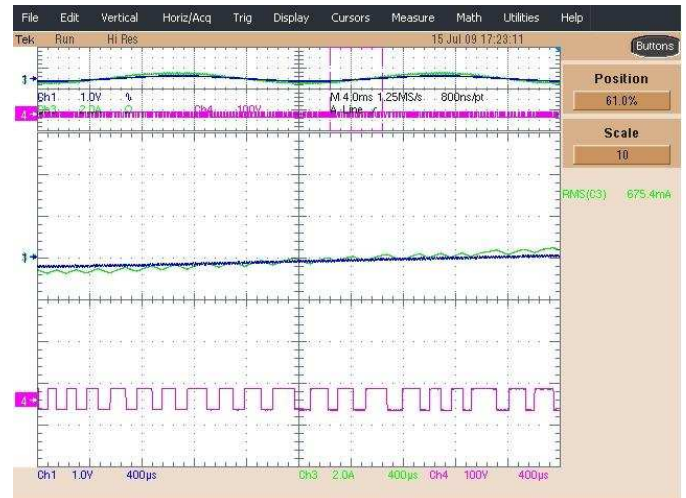
Las señales que se representan en las capturas tomadas por el osciloscopio son las siguientes:

- En **azul**, se muestra la corriente de referencia, obtenida tras la digitalización de la tensión de red sensada y pasada a través de un módulo de conversión D/A, por lo que se representa como una tensión en lugar de una corriente.
- En **verde**, se representa la corriente de salida del inversor.
- En **magenta**, se puede observar la tensión de salida del propio puente inversor.

A continuación se representan los diversos ensayos en cortocircuito:



Ensayo de cortocircuito con H1, R1, $V_{panel}=25\text{ Vdc}$, $I_{panel}=0.06\text{ Adc}$.



Ensayo de cortocircuito con H1, R2, $V_{panel}=30\text{ Vdc}$, $I_{panel}=0.20\text{ Adc}$.



Ensayo de cortocircuito con H2, R1, $V_{panel}=25\text{ Vdc}$, $I_{panel}=0.11\text{ Adc}$.

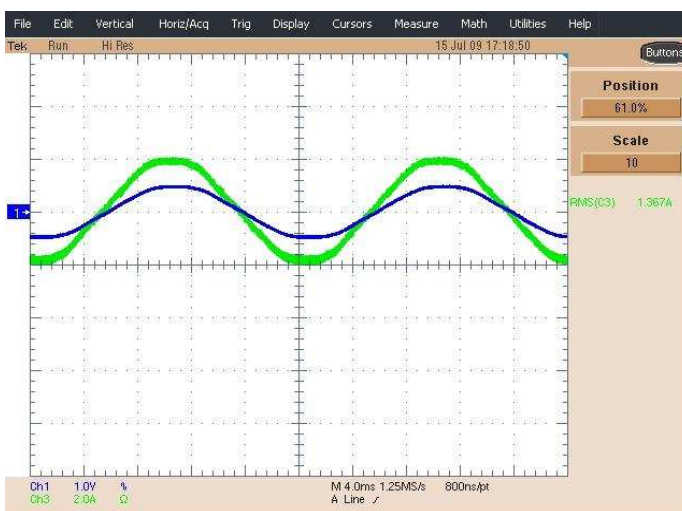




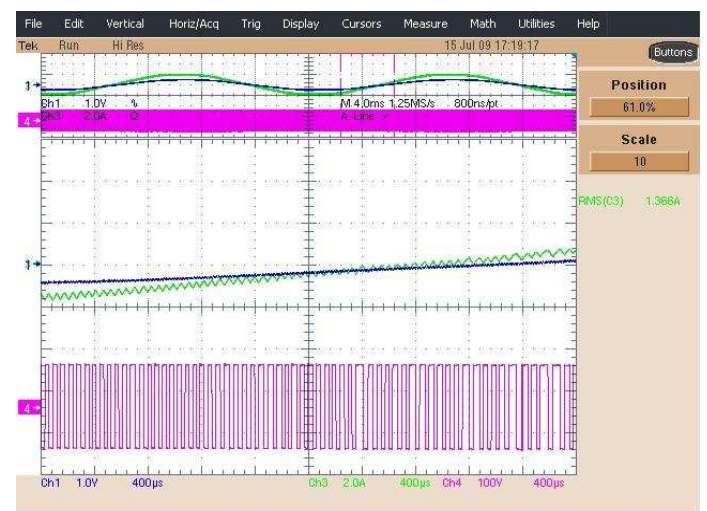
Ensayo de cortocircuito con H2, R2, $V_{panel}=30\text{ Vdc}$, $I_{panel}=0.19\text{ Adc}$.

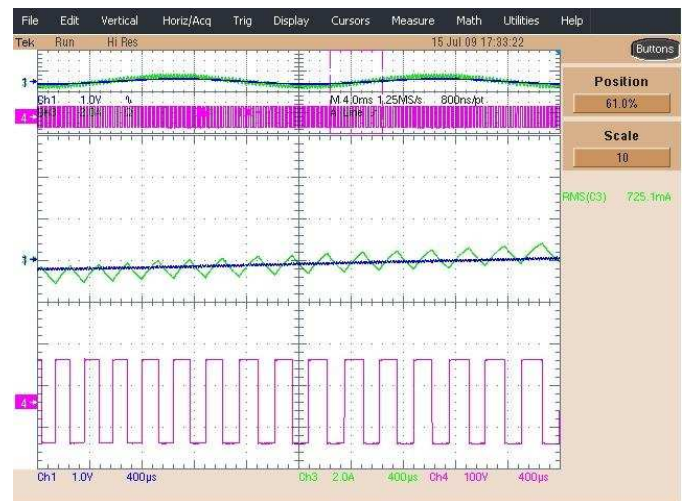
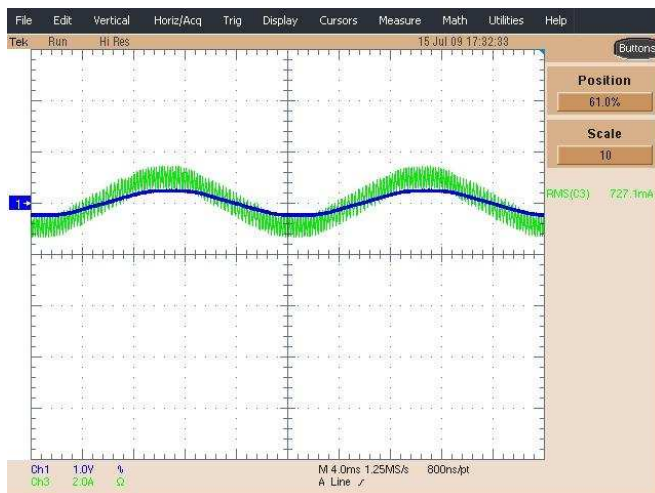


Ensayo de cortocircuito con H1, R1, $V_{panel}=100\text{ Vdc}$, $I_{panel}=0.05\text{ Adc}$.

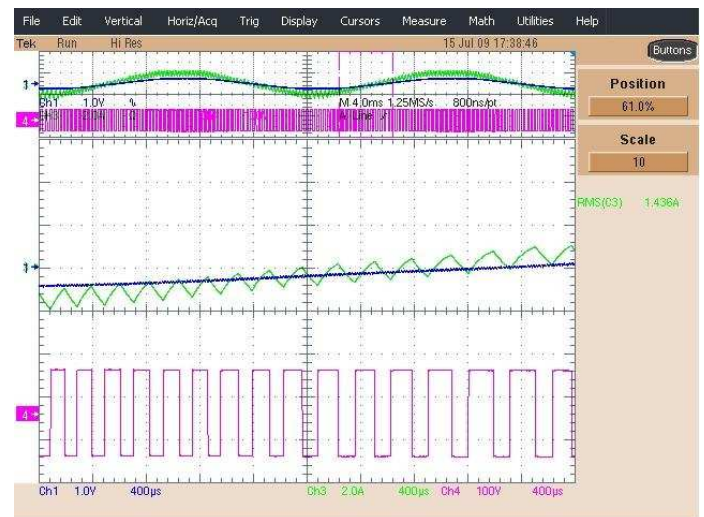
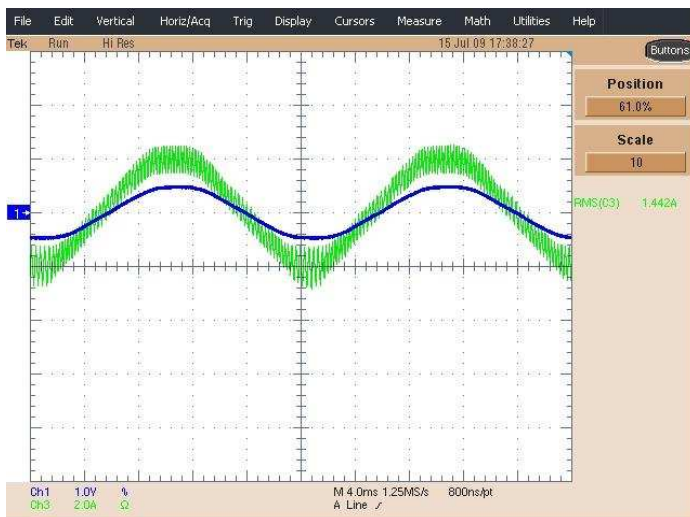


Ensayo de cortocircuito con H1, R2, $V_{panel}=100\text{ Vdc}$, $I_{panel}=0.10\text{ Adc}$.





Ensayo de cortocircuito con H2, R1, $V_{panel}=100\text{ Vdc}$, $I_{panel}=0.07\text{ Adc}$.



Ensayo de cortocircuito con H2, R2, $V_{panel}=100\text{ Vdc}$, $I_{panel}=0.12\text{ Adc}$.

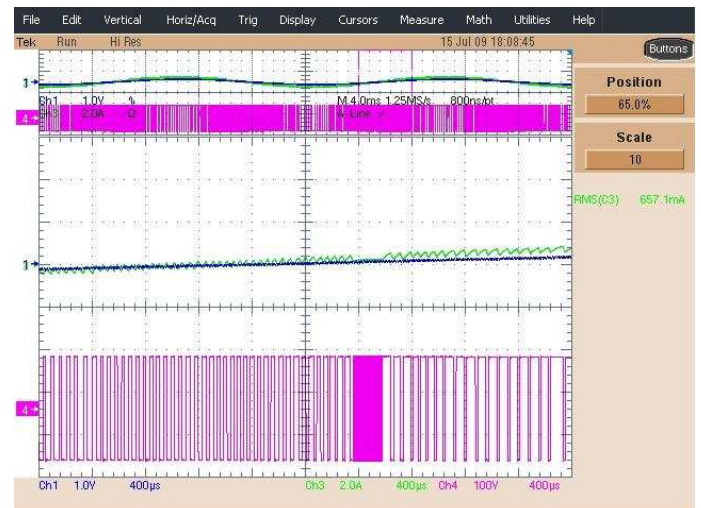
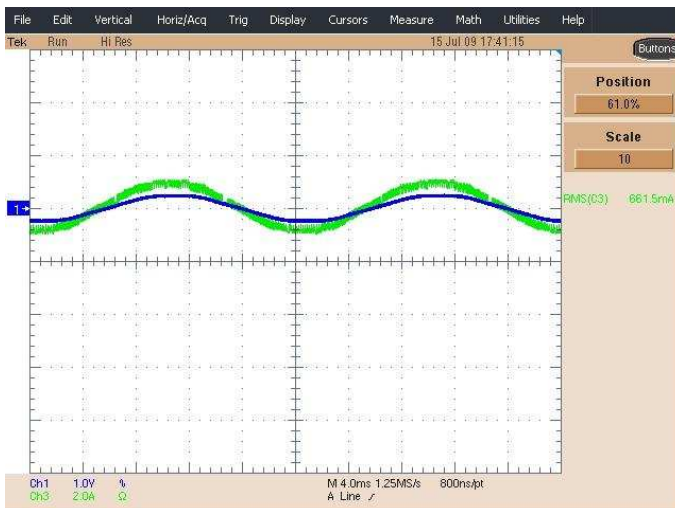
B. ENSAYOS CON CARGA RESISTIVA

Estas pruebas se realizan con una carga resistiva a la salida del inversor de valor $R = 106 \, \Omega$, en serie con la inductancia L_2 de conexión a red.

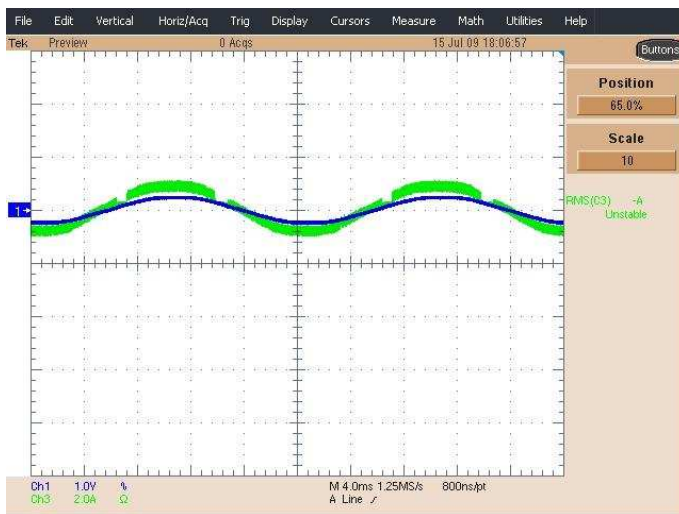
Las señales que se representan en las capturas tomadas por el osciloscopio son las siguientes:

- En **azul**, se muestra la corriente de referencia, obtenida tras la digitalización de la tensión de red sensada y pasada a través de un módulo de conversión D/A, por lo que se representa como una tensión en lugar de una corriente.
- En **verde**, se representa la corriente de salida del inversor.
- En **magenta**, se puede observar la tensión de salida del propio puente inversor.

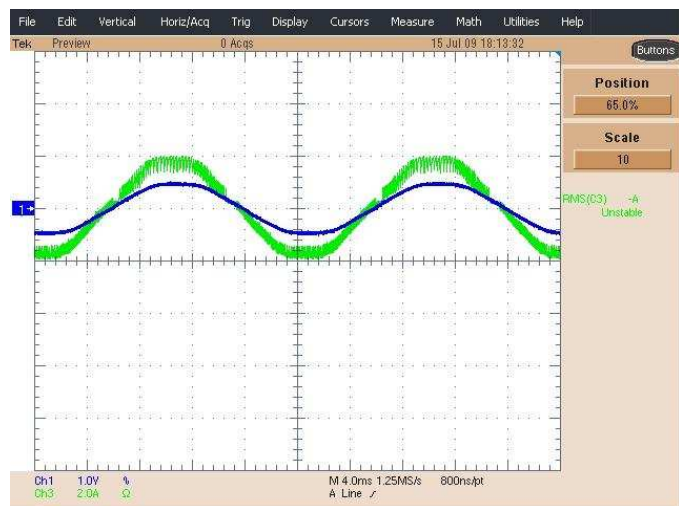
Las pruebas realizadas con la carga resistiva conectada a la salida del inversor son las siguientes:



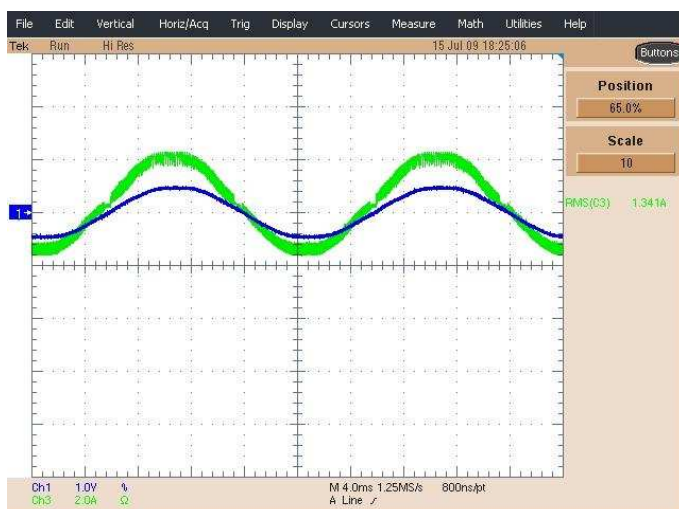
Ensayo con carga resistiva $R = 106 \, \Omega$, con $H1$, $R1$, $V_{panel} = 120 \, V_{dc}$, $I_{panel} = 0.45 \, A_{dc}$.



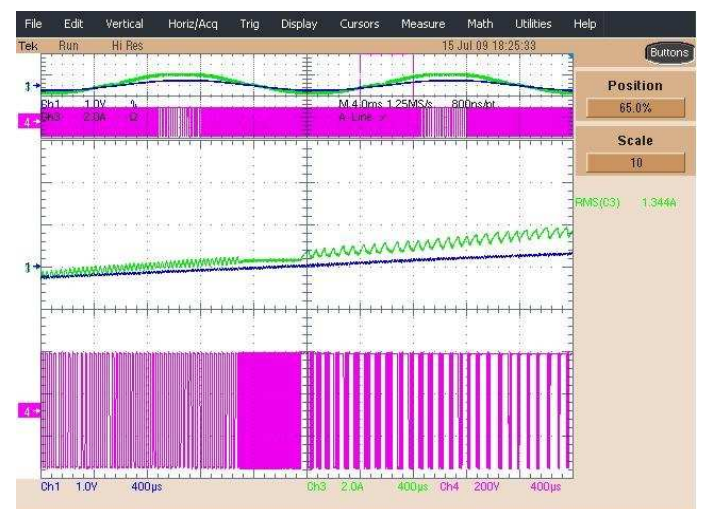
Ensayo con carga resistiva $R = 106 \Omega$, con $H1$, $R1$, $V_{panel} = 180 \text{ Vdc}$, $I_{panel} = 0.31 \text{ Adc}$.

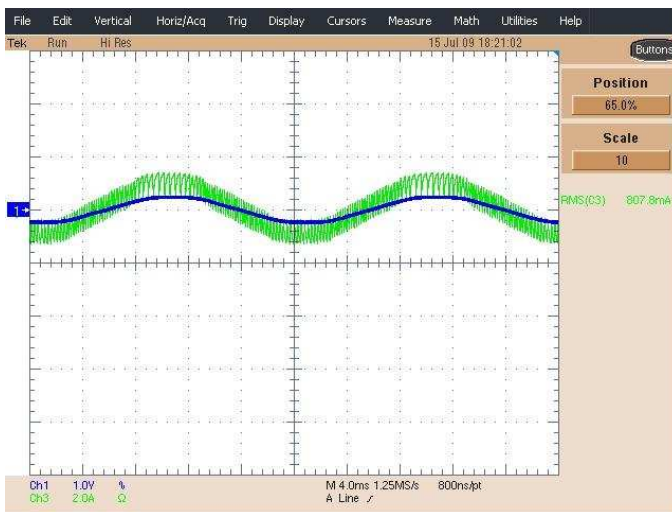


Ensayo con carga resistiva $R = 106 \Omega$, con $H1$, $R2$, $V_{panel} = 180 \text{ Vdc}$, $I_{panel} = 0.91 \text{ Adc}$.

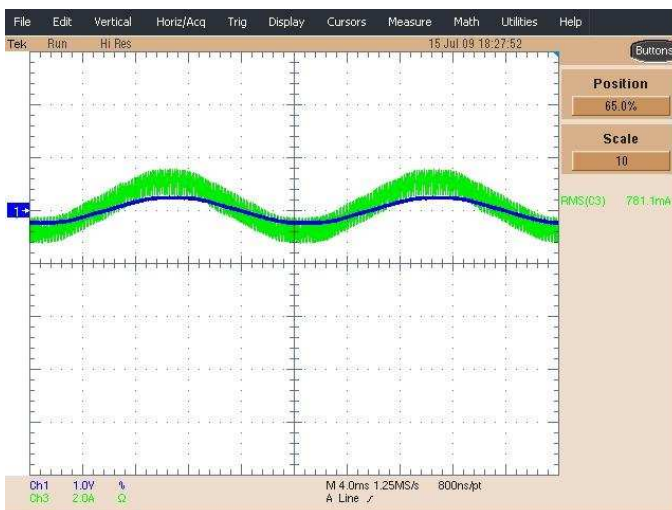
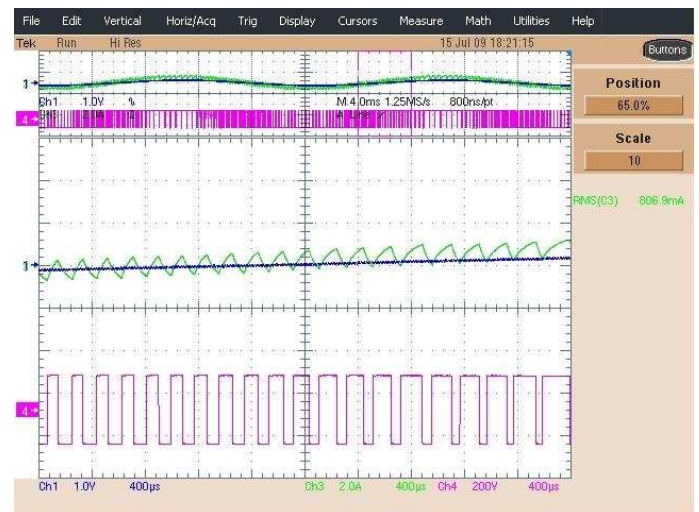


Ensayo con carga resistiva $R = 106 \Omega$, con $H1$, $R2$, $V_{panel} = 270 \text{ Vdc}$, $I_{panel} = 0.85 \text{ Adc}$.

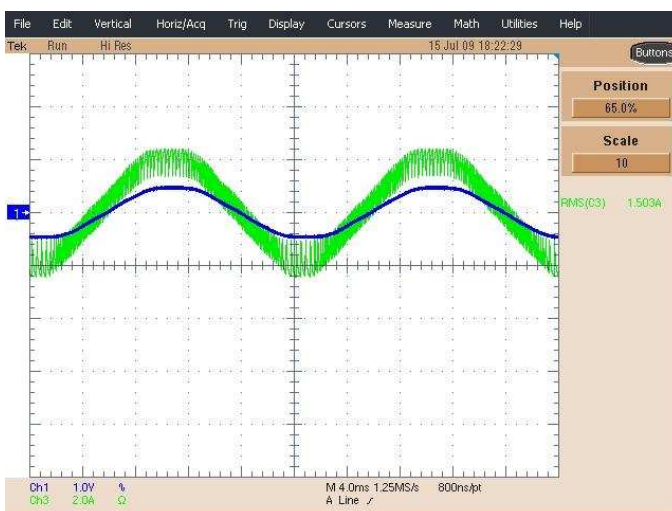
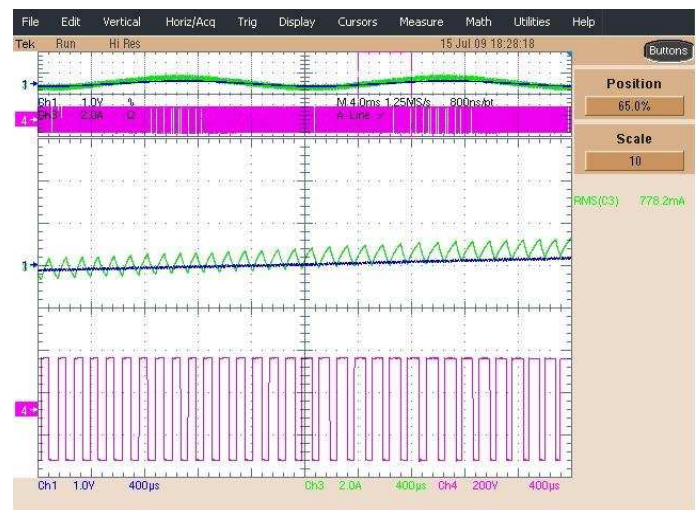




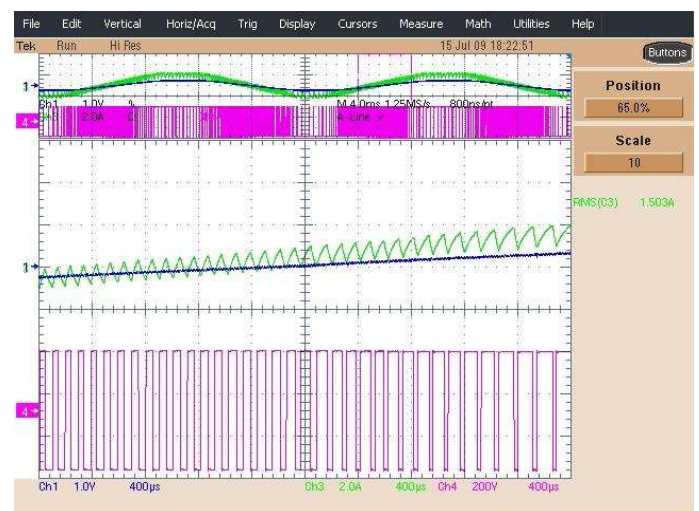
Ensayo con carga resistiva $R = 106 \Omega$, con $H2, R1, V_{panel} = 160 V_{dc}, I_{panel} = 0.48 A_{dc}$.



Ensayo con carga resistiva $R = 106 \Omega$, con $H2, R1, V_{panel} = 240 V_{dc}, I_{panel} = 0.33 A_{dc}$.



Ensayo con carga resistiva $R = 106 \Omega$, con $H2, R2, V_{panel} = 280 V_{dc}, I_{panel} = 0.97 A_{dc}$.



C. ENSAYOS DE CONEXIÓN A RED

Los ensayos de conexión a red se realizan con un transformador de aislamiento a la salida del inversor y un autotransformador conectado a la red que permite regular la tensión de la misma. Estas pruebas se han realizado tanto con la bobina de conexión a red L_2 como con un filtro LCL que permite reducir el contenido armónico de la corriente inyectada.

En la siguiente tabla, se representan los principales resultados obtenidos de los ensayos realizados con el inversor conectado a red, siendo las condiciones de funcionamiento comunes: La referencia de corriente para el lazo de corriente en este caso es de 1 A de pico (R1) y la banda de histéresis es la más ancha de las 2 seleccionables (H2).

	Referencia de corriente R1 (1 A _{PICO}), Banda de histéresis ancha (H2)					
L₂	V_{PANEL} (V_{DC})	70	92	123	151	179
	I_{PANEL} (I_{DC})	0.36	0.40	0.41	0.42	0.41
	P_{PANEL} (W)	25.2	36.8	50.43	63.42	73.39
	V_{RED} (V_{ef})	40	60	76	93.67	110
	P_{AC} (W)	20	30	40	48.47	52.33
	η (%)	79.37	81.52	79.32	76.40	71.30
Filtro LCL	V_{PANEL} (V_{DC})	71	128	183	186	
	I_{PANEL} (I_{DC})	0.36	0.40	0.35	0.41	
	P_{PANEL} (W)	25.56	51.2	64.05	76.26	
	V_{RED} (V_{ef})	39.4	75	111.8	111.3	
	P_{AC} (W)	22	40	56.76	56.26	
	η (%)	86.1	78.12	88.62	73.77	

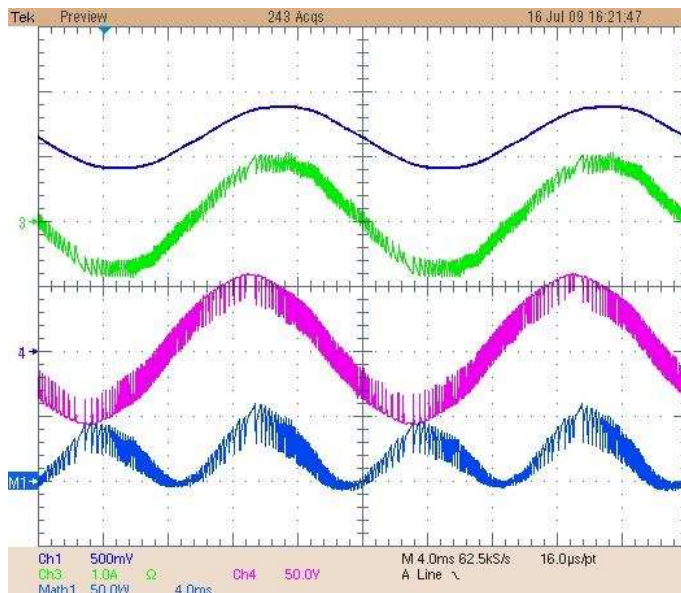
En la tabla que se muestra a continuación se representan los resultados más significativos de realizar los ensayos de conexión a red con las siguientes condiciones de funcionamiento: La referencia de corriente para el lazo de corriente en este caso es

de 2 A de pico (R2) y la banda de histéresis es la más ancha de las 2 seleccionables (H2).

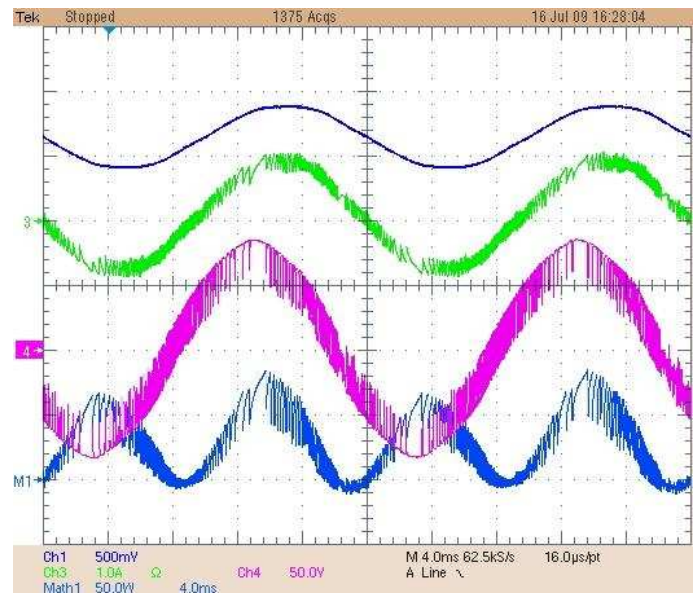
	Referencia de corriente R2 (2 A _{PICO}), Banda de histéresis ancha (H2)					
L₂	V_{PANEL} (V_{DC})	76	97	124	153	177
	I_{PANEL} (I_{DC})	0.77	0.83	0.84	0.86	0.86
	P_{PANEL} (W)	58.52	80.51	104.16	131.58	152.22
	V_{RED} (V_{ef})	42	59	76.76	96.2	112.7
	P_{AC} (W)	45	63	81	100.3	113.7
	η (%)	77.32	78.25	77.76	76.23	74.69
Filtro LCL	V_{PANEL} (V_{DC})	71	129	190	186	315
	I_{PANEL} (I_{DC})	0.78	0.81	0.82	0.82	0.84
	P_{PANEL} (W)	55.38	104.49	155.8	152.52	264.6
	V_{RED} (V_{ef})	39	76.66	112.6	113.4	210
	P_{AC} (W)	45	79.84	122.3	124.3	200
	η (%)	81.26	76.40	78.50	81.50	75.59

Las principales medidas correspondientes a los resultados representados en estas tablas se pueden observar en las capturas tomadas con el osciloscopio que se muestran a continuación. Las señales representadas en estas gráficas son:

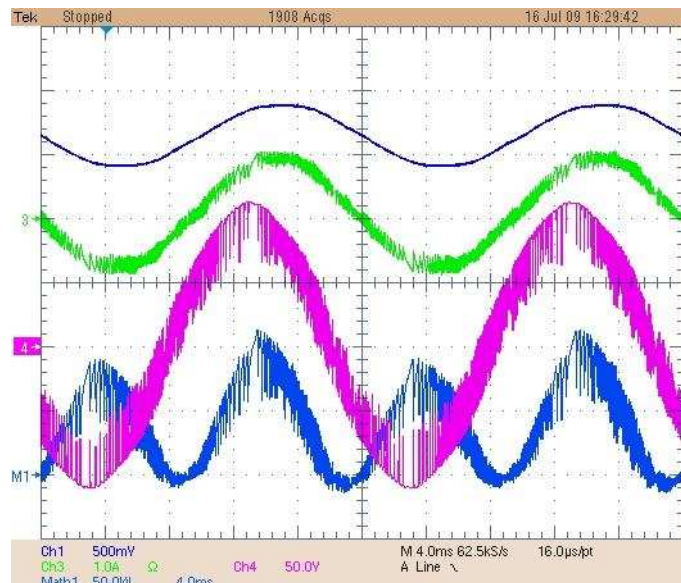
- En **azul oscuro**, se muestra la corriente de referencia, obtenida tras la digitalización de la tensión de red sensada y pasada a través de un módulo de conversión D/A, por lo que se representa como una tensión en lugar de una corriente.
- En **rojo**, la corriente de salida sensada y digitalizada, pasada por el conversor D/A.
- En **verde**, se representa la corriente de salida del inversor.
- En **magenta**, se puede observar la tensión de red.
- En **azul claro**, se representa la potencia instantánea entregada por el puente inversor, obtenida como el producto de la tensión de red y la corriente inyectada.



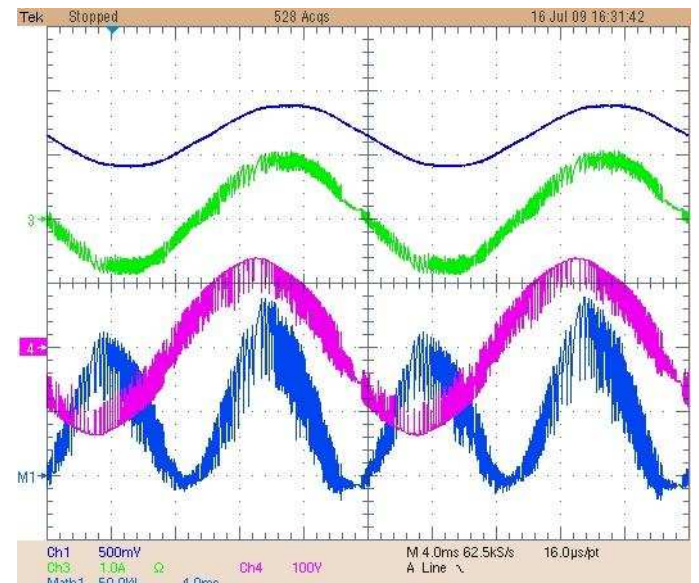
Ensayo de conexión a red con inductancia de conexión L_2 . Condiciones de funcionamiento: R1, H2, $V_{PANEL}=70$ V_{DC} .



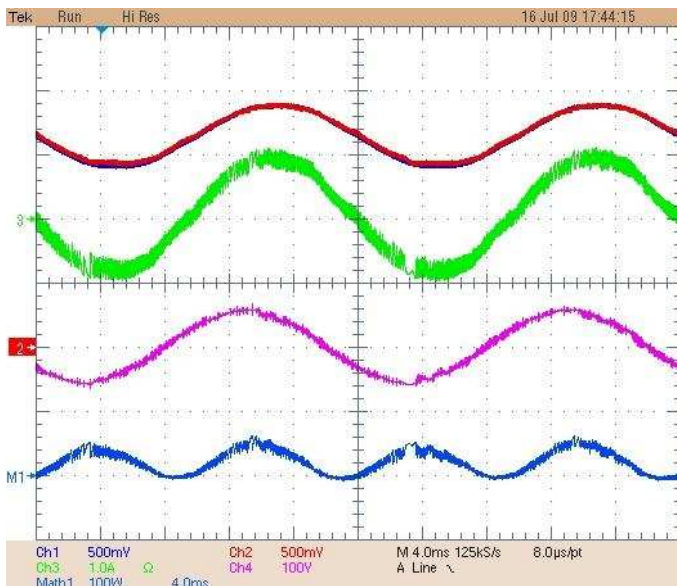
Ensayo de conexión a red con inductancia de conexión L_2 . Condiciones de funcionamiento: R1, H2, $V_{PANEL}=92$ V_{DC} .



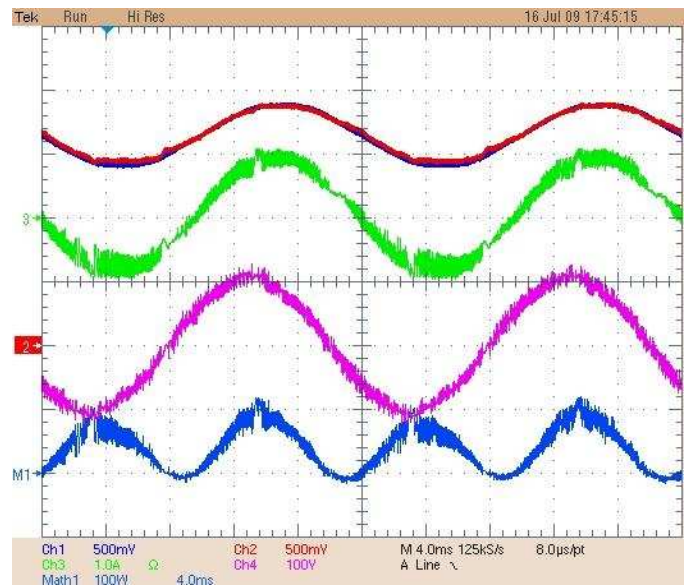
Ensayo de conexión a red con inductancia de conexión L_2 . Condiciones de funcionamiento: R1, H2, $V_{PANEL}=123$ V_{DC} .



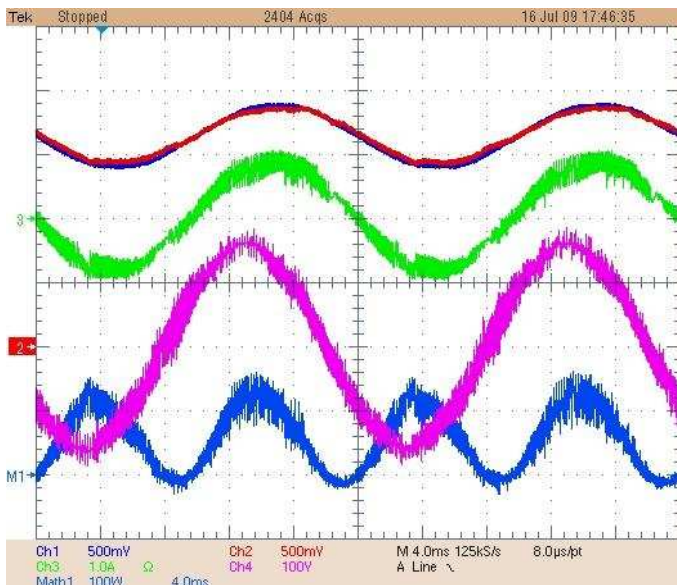
Ensayo de conexión a red con inductancia de conexión L_2 . Condiciones de funcionamiento: R1, H2, $V_{PANEL}=151$ V_{DC} .



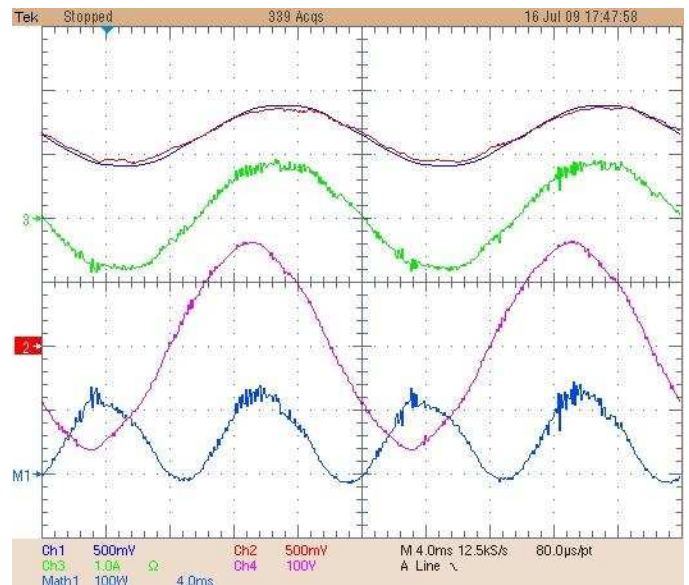
Ensayo de conexión a red con filtro LCL. Condiciones de funcionamiento: $R1, H2, V_{PANEL}=71 V_{DC}$.



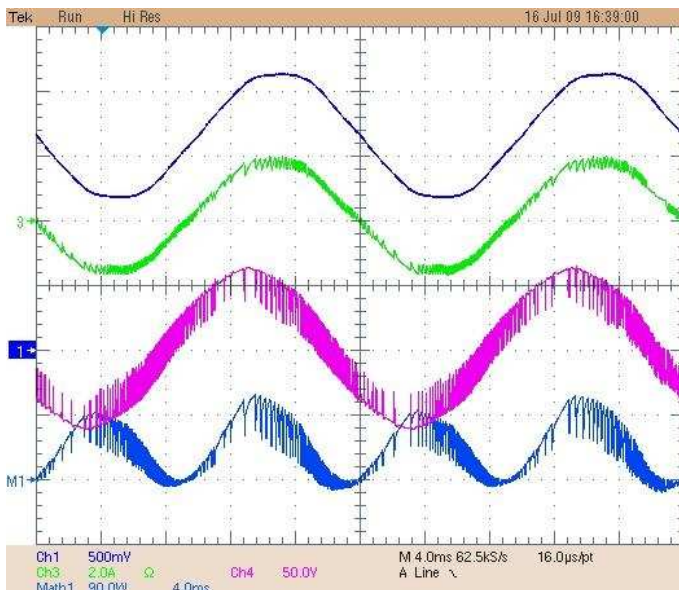
Ensayo de conexión a red con filtro LCL. Condiciones de funcionamiento: $R1, H2, V_{PANEL}=128 V_{DC}$.



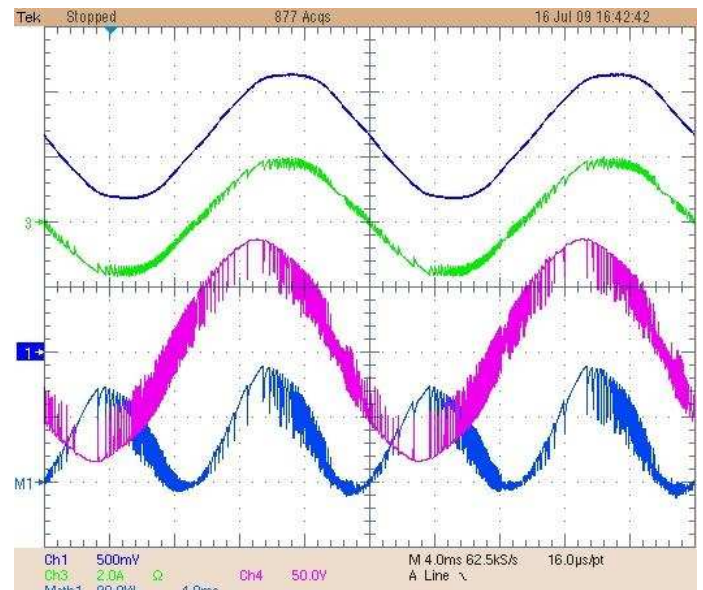
Ensayo de conexión a red con filtro LCL. Condiciones de funcionamiento: $R1, H2, V_{PANEL}=183 V_{DC}$.



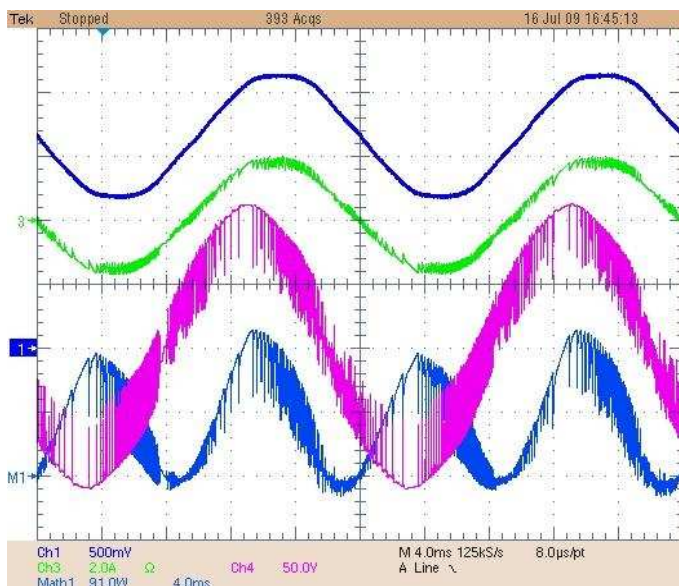
Ensayo de conexión a red con filtro LCL. Condiciones de funcionamiento: $R1, H2, V_{PANEL}=186 V_{DC}$. Baja resolución.



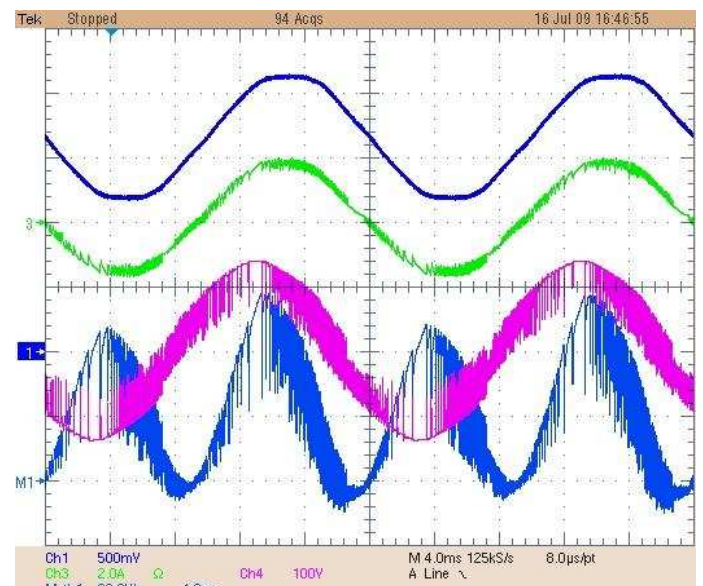
Ensayo de conexión a red con inductancia de conexión L_2 . Condiciones de funcionamiento: R2, H2, $V_{PANEL}=76$ V_{DC} .



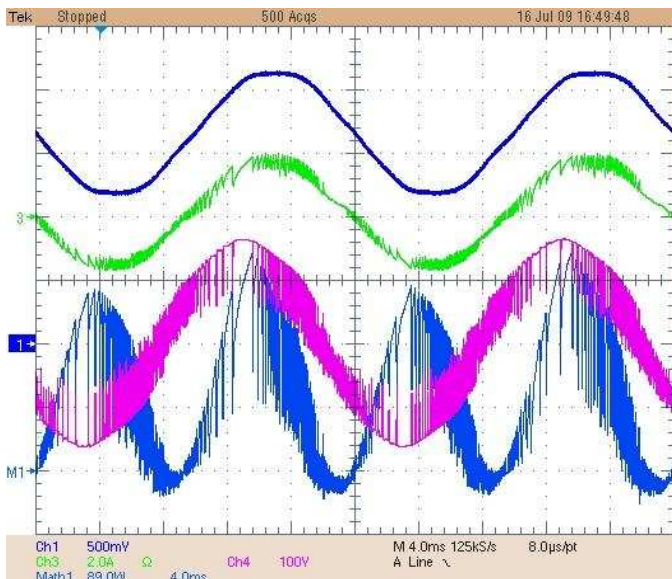
Ensayo de conexión a red con inductancia de conexión L_2 . Condiciones de funcionamiento: R2, H2, $V_{PANEL}=97$ V_{DC} .



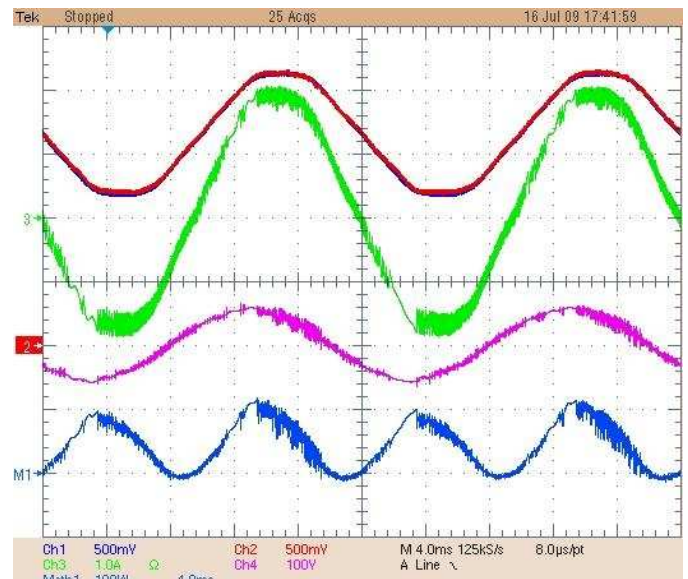
Ensayo de conexión a red con inductancia de conexión L_2 . Condiciones de funcionamiento: R2, H2, $V_{PANEL}=124$ V_{DC} .



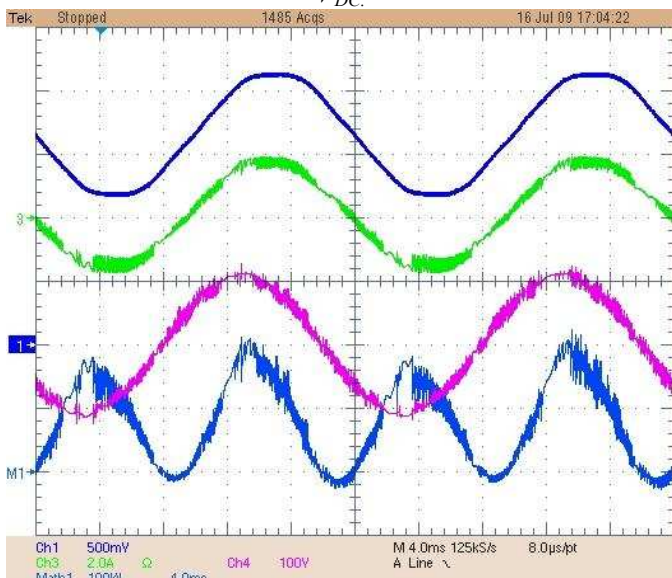
Ensayo de conexión a red con inductancia de conexión L_2 . Condiciones de funcionamiento: R2, H2, $V_{PANEL}=153$ V_{DC} .



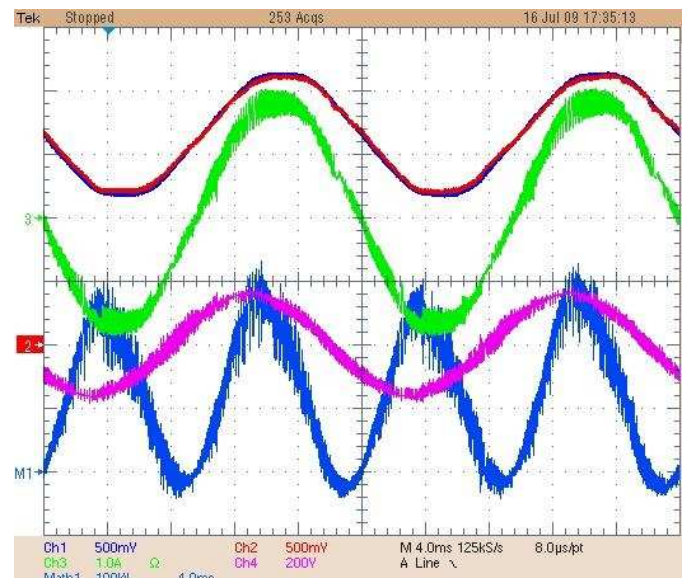
Ensayo de conexión a red con inductancia de conexión L_2 . Condiciones de funcionamiento: R_2 , H_2 , $V_{PANEL}=177 V_{DC}$.



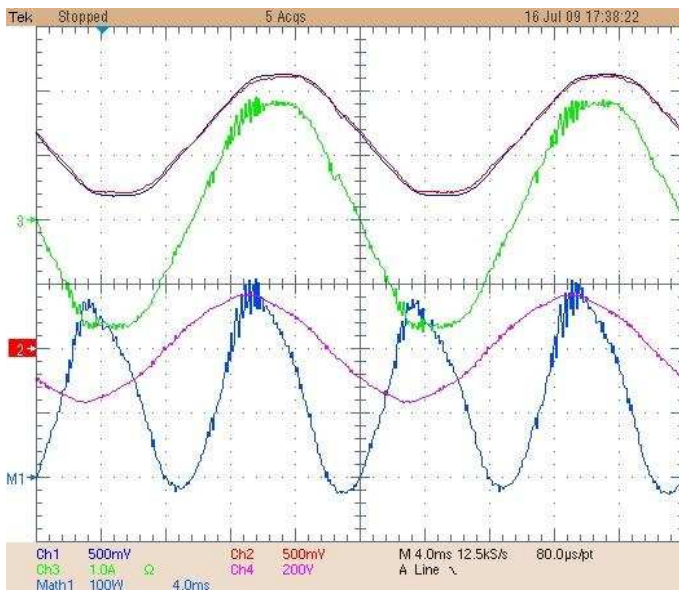
Ensayo de conexión a red con filtro LCL. Condiciones de funcionamiento: R_2 , H_2 , $V_{PANEL}=71 V_{DC}$.



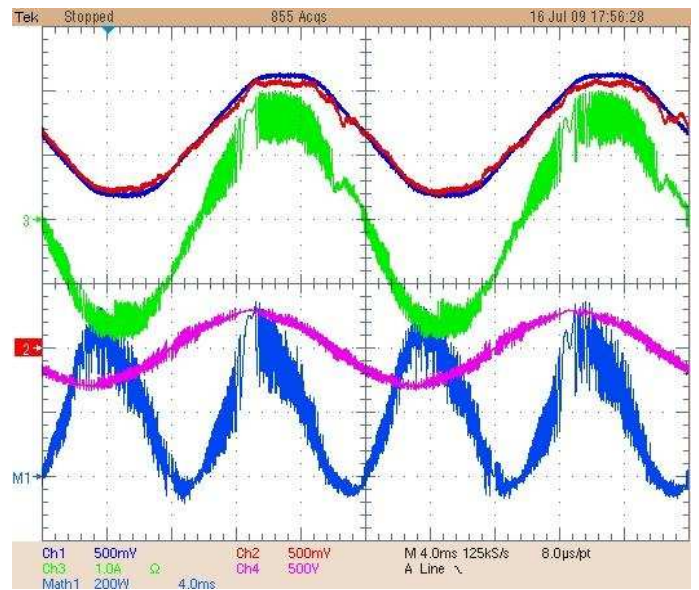
Ensayo de conexión a red con filtro LCL. Condiciones de funcionamiento: R_2 , H_2 , $V_{PANEL}=129 V_{DC}$.



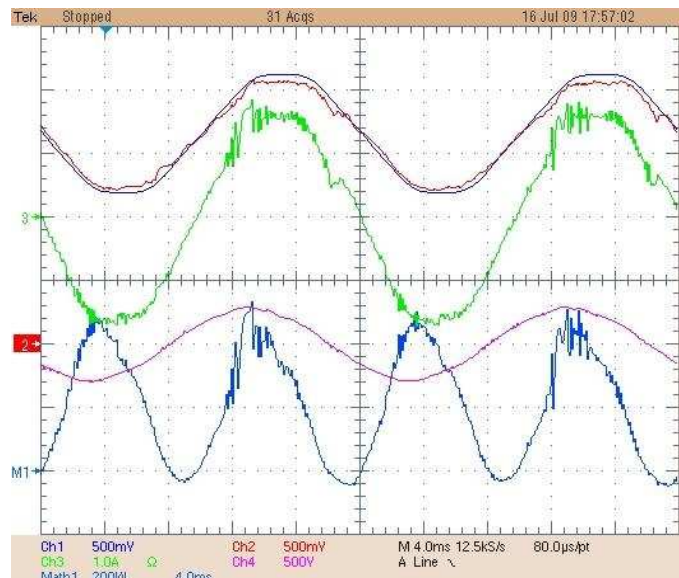
Ensayo de conexión a red con filtro LCL. Condiciones de funcionamiento: R_2 , H_2 , $V_{PANEL}=190 V_{DC}$.



Ensayo de conexión a red con filtro LCL. Condiciones de funcionamiento: R2, H2, $V_{PANEL}=186 V_{DC}$. Baja resolución.

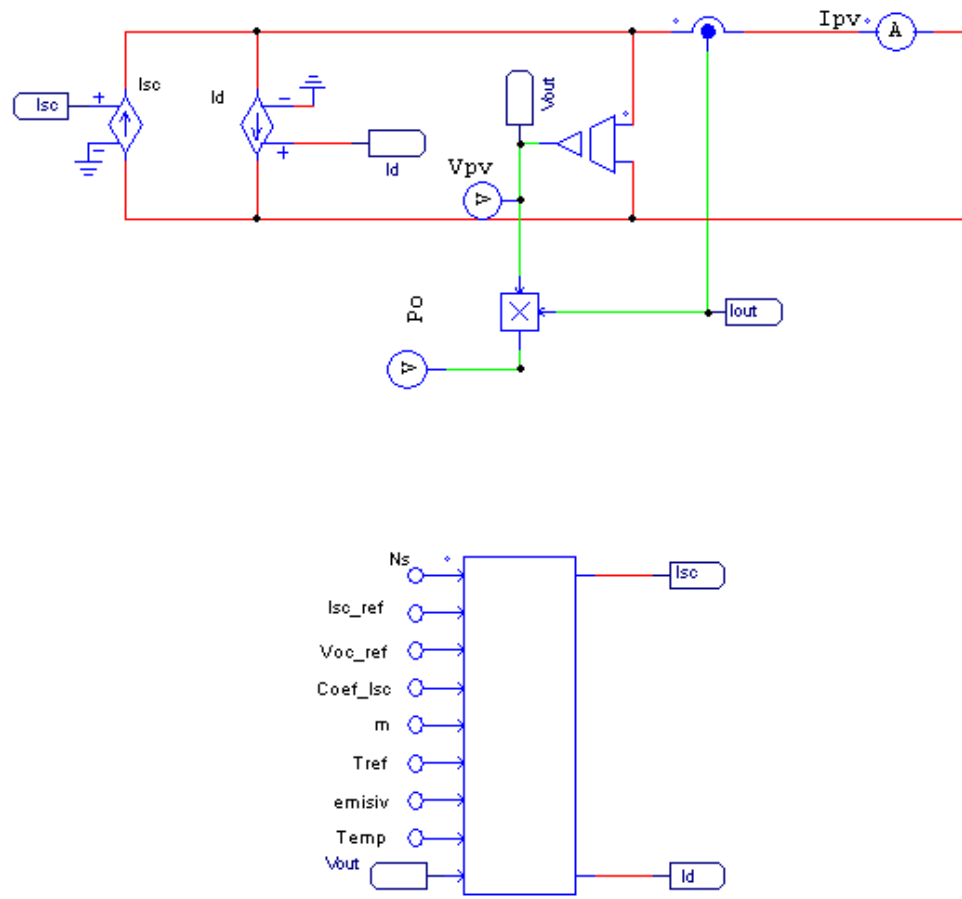


Ensayo de conexión a red con filtro LCL. Condiciones de funcionamiento: R2, H2, $V_{PANEL}=315 V_{DC}$.



Ensayo de conexión a red con filtro LCL. Condiciones de funcionamiento: R2, H2, $V_{PANEL}=315 V_{DC}$. Baja resolución.

ANEXO X. MODELO DEL PANEL FOTOVOLTAICO PARA SIMULACIONES CON PSIM



Este modelo ha sido realizado por *Carlos David Martínez Nieto*



ANEXO XI. LISTA DE COMPONENTES EMPLEADOS

En la siguiente lista se detalla el modelo de todos los componentes empleados en el diseño de los principales bloques de este proyecto (etapa de potencia, bloque de sensado, acondicionamiento, fuentes auxiliares y drivers).

COMPONENTES
Transductor LV25-P
Transductor LTS 25-NP
MOSFET IRF740
Driver IR2110
Tarjeta de evaluación de FPGA, modelo Basys
Convertidor ADS7887
Convertidor CA/CC, modelo TOM 12215
Convertidor CC/CC, modelo TEL 5-1211
Amplificador operacional dual LMV712
SN74LVC2G17 Dual Schmitt Trigger Buffer